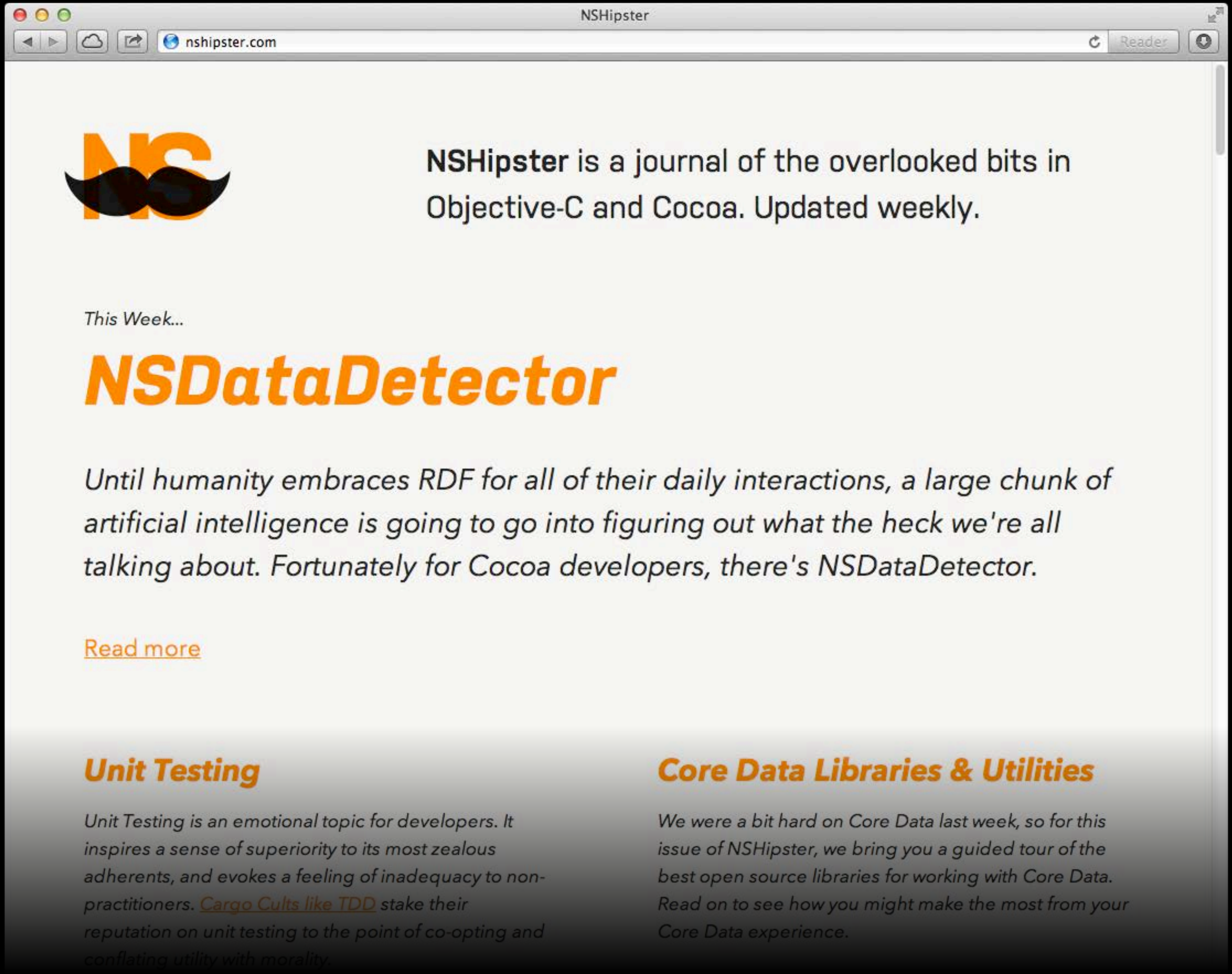
Hidden Gems in Cocoa and Cocoa Touch

Session 228

Scott Stevenson
Software Engineer

NSHipster.com

Mattt Thompson



NSHipster is a journal of the overlooked bits in Objective-C and Cocoa. Updated weekly.

This Week...

NSDataDetector

Until humanity embraces RDF for all of their daily interactions, a large chunk of artificial intelligence is going to go into figuring out what the heck we're all talking about. Fortunately for Cocoa developers, there's NSDataDetector.

Read more

Unit Testing

Unit Testing is an emotional topic for developers. It inspires a sense of superiority to its most zealous adherents, and evokes a feeling of inadequacy to non-practitioners. Cargo Cults like TDD stake their reputation on unit testing to the point of co-opting and conflating utility with morality.

Core Data Libraries & Utilities

We were a bit hard on Core Data last week, so for this issue of NSHipster, we bring you a guided tour of the best open source libraries for working with Core Data. Read on to see how you might make the most from your Core Data experience.

AppKit
UIKit
Objective-C
Core Data
Foundation
Xcode
Core Animation

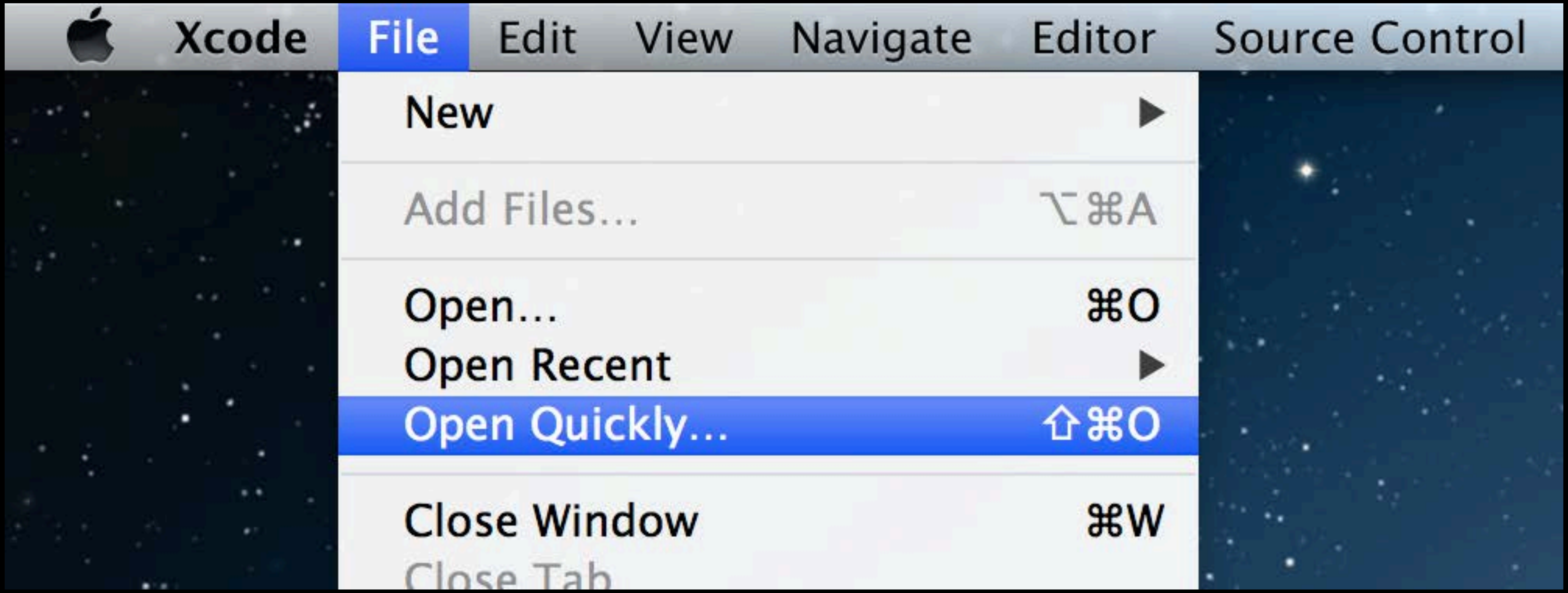
Xcode

Editing and debugging

Xcode

Open Quickly (Command-Shift-O)

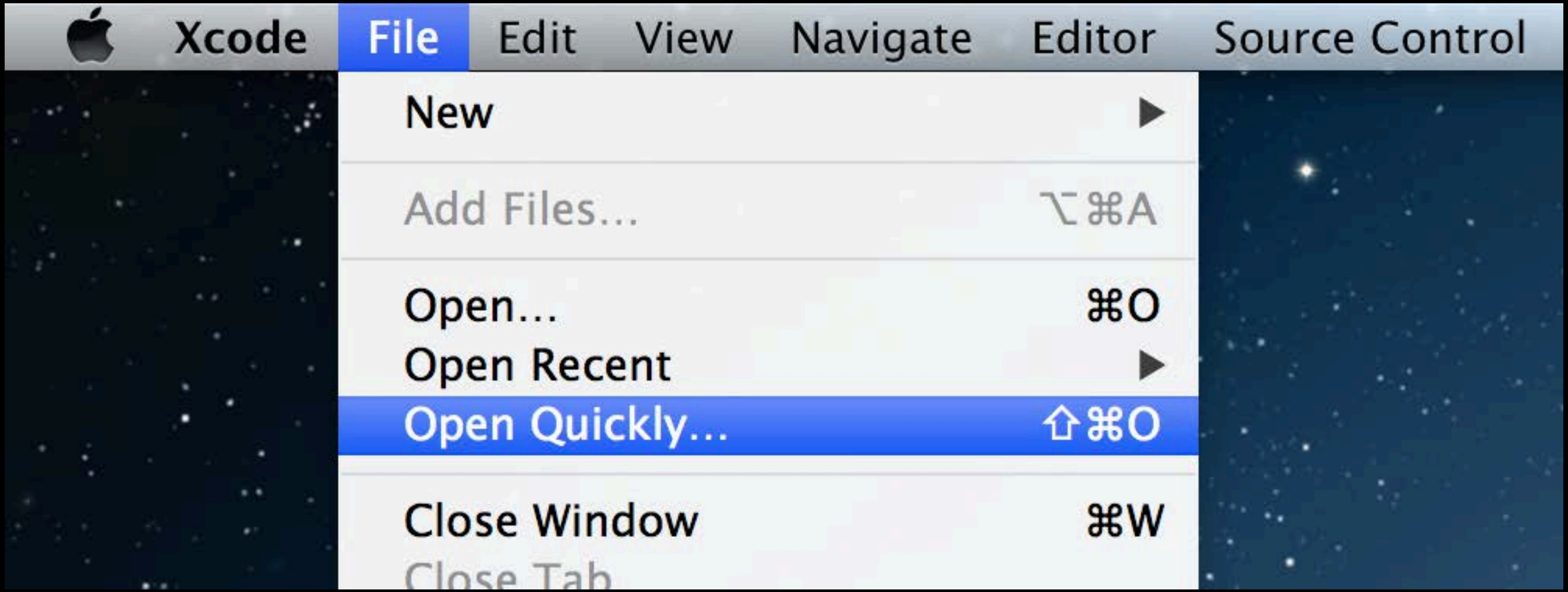
Tip 1



Xcode

Open Quickly (Command-Shift-O)

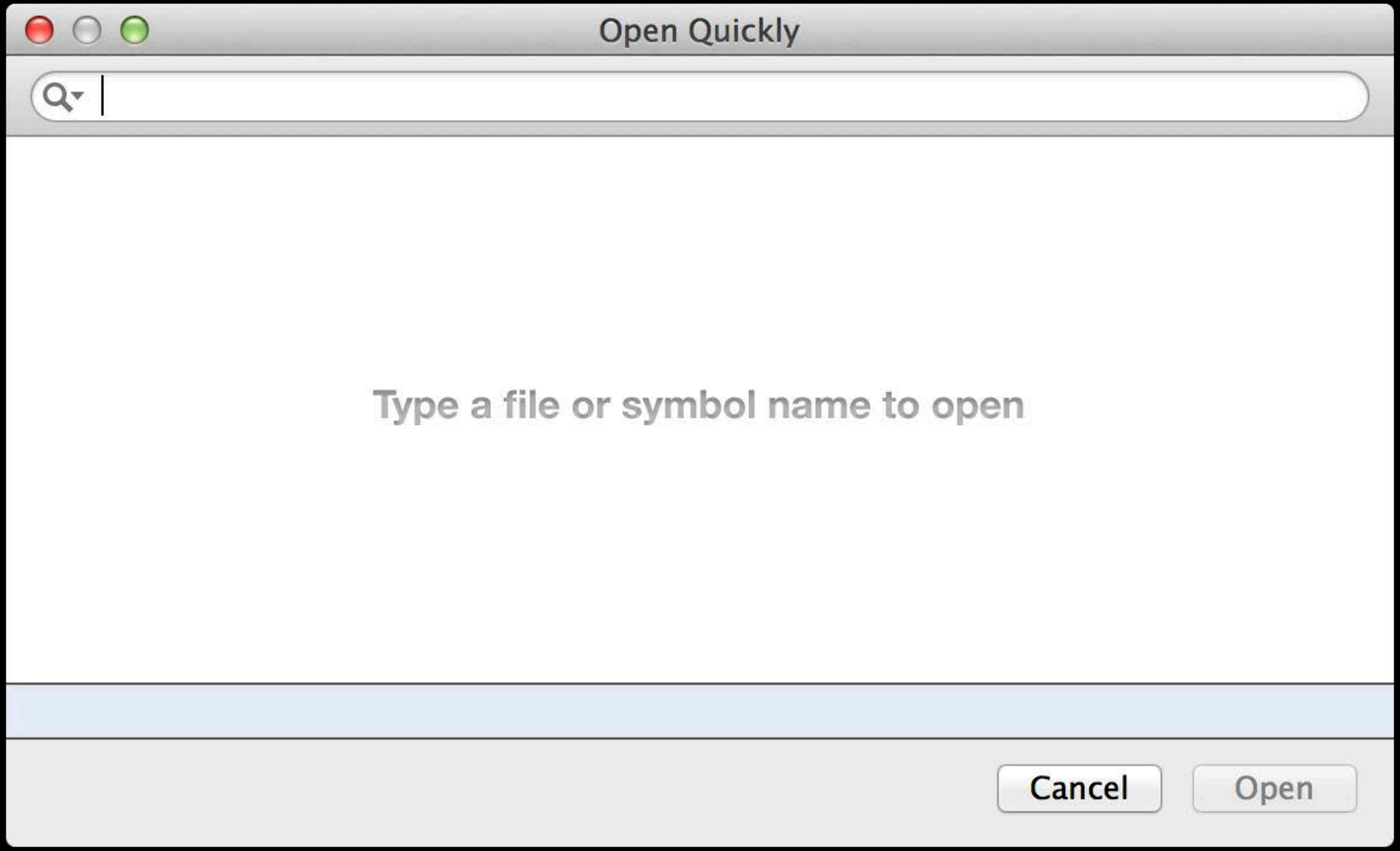
Tip 1



Xcode

Open Quickly (Command-Shift-O)

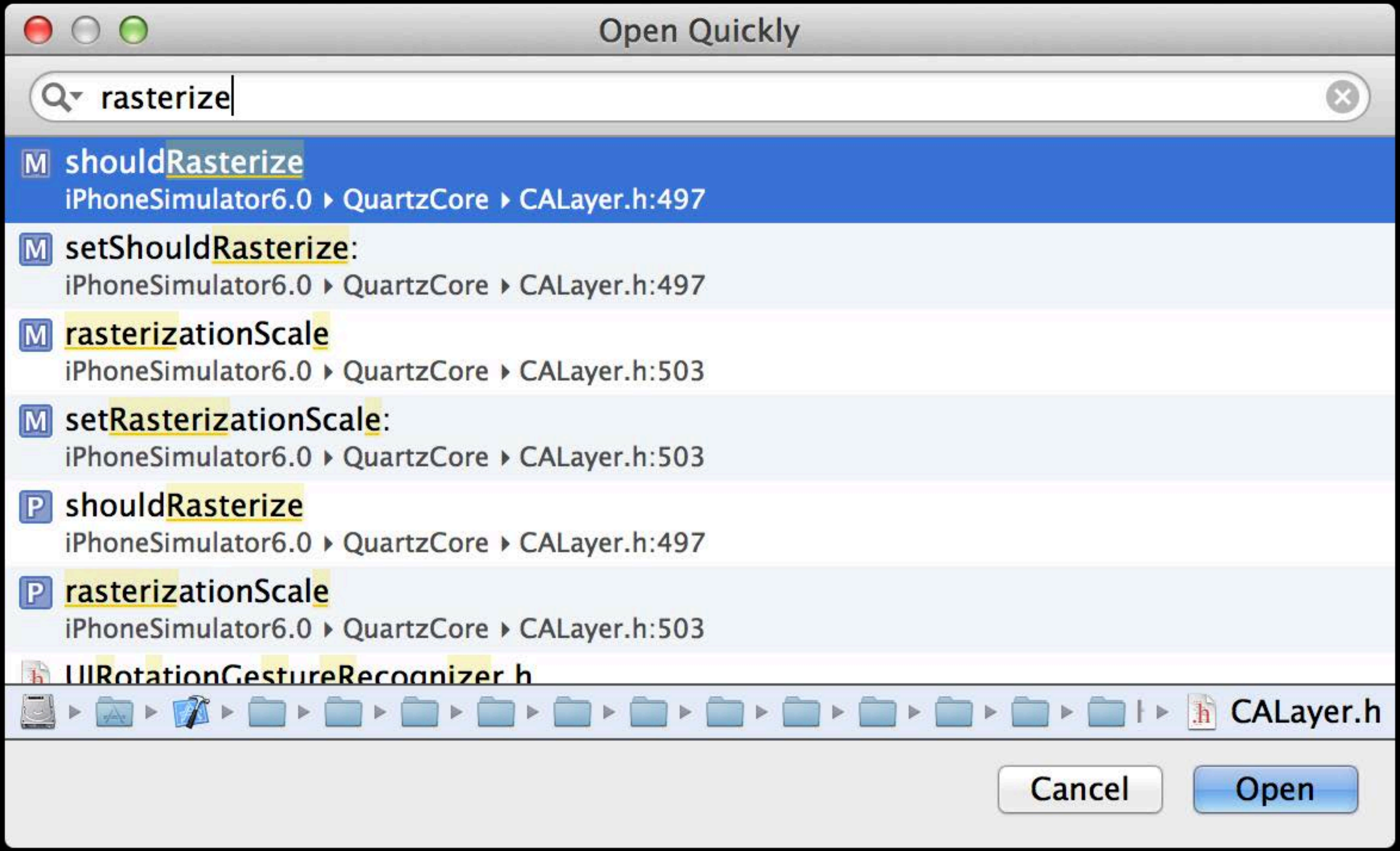
Tip
1



Xcode

Open Quickly (Command-Shift-O)

Tip
1


Open Quickly

rasterize

M shouldRasterize
iPhoneSimulator6.0 ▸ QuartzCore ▸ CALayer.h:497

M setShouldRasterize:
iPhoneSimulator6.0 ▸ QuartzCore ▸ CALayer.h:497

M rasterizationScale
iPhoneSimulator6.0 ▸ QuartzCore ▸ CALayer.h:503

M setRasterizationScale:
iPhoneSimulator6.0 ▸ QuartzCore ▸ CALayer.h:503

P shouldRasterize
iPhoneSimulator6.0 ▸ QuartzCore ▸ CALayer.h:497

P rasterizationScale
iPhoneSimulator6.0 ▸ QuartzCore ▸ CALayer.h:503

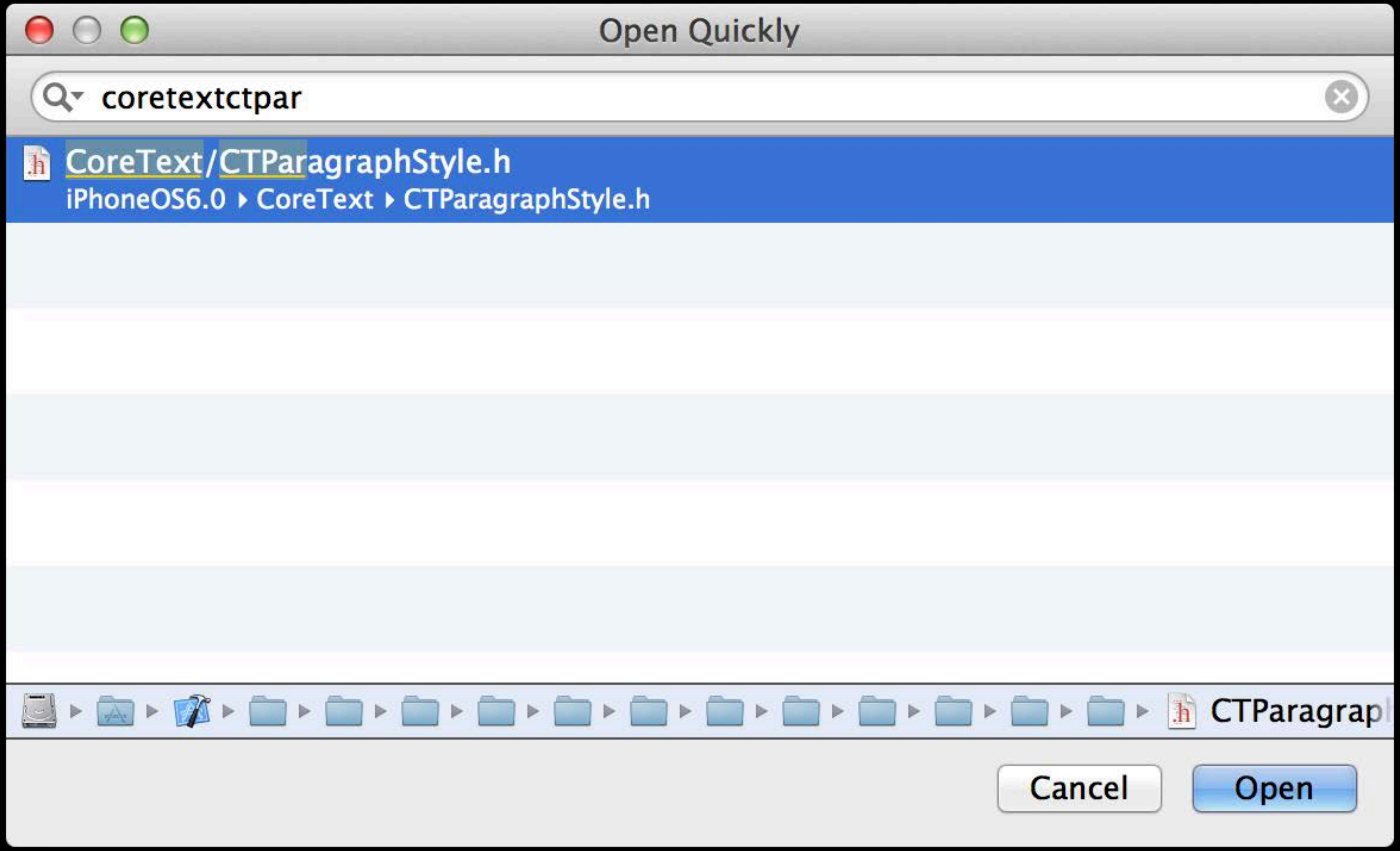
UIRotationGestureRecognizer.h

CALayer.h

Xcode

Open Quickly (Command-Shift-O)

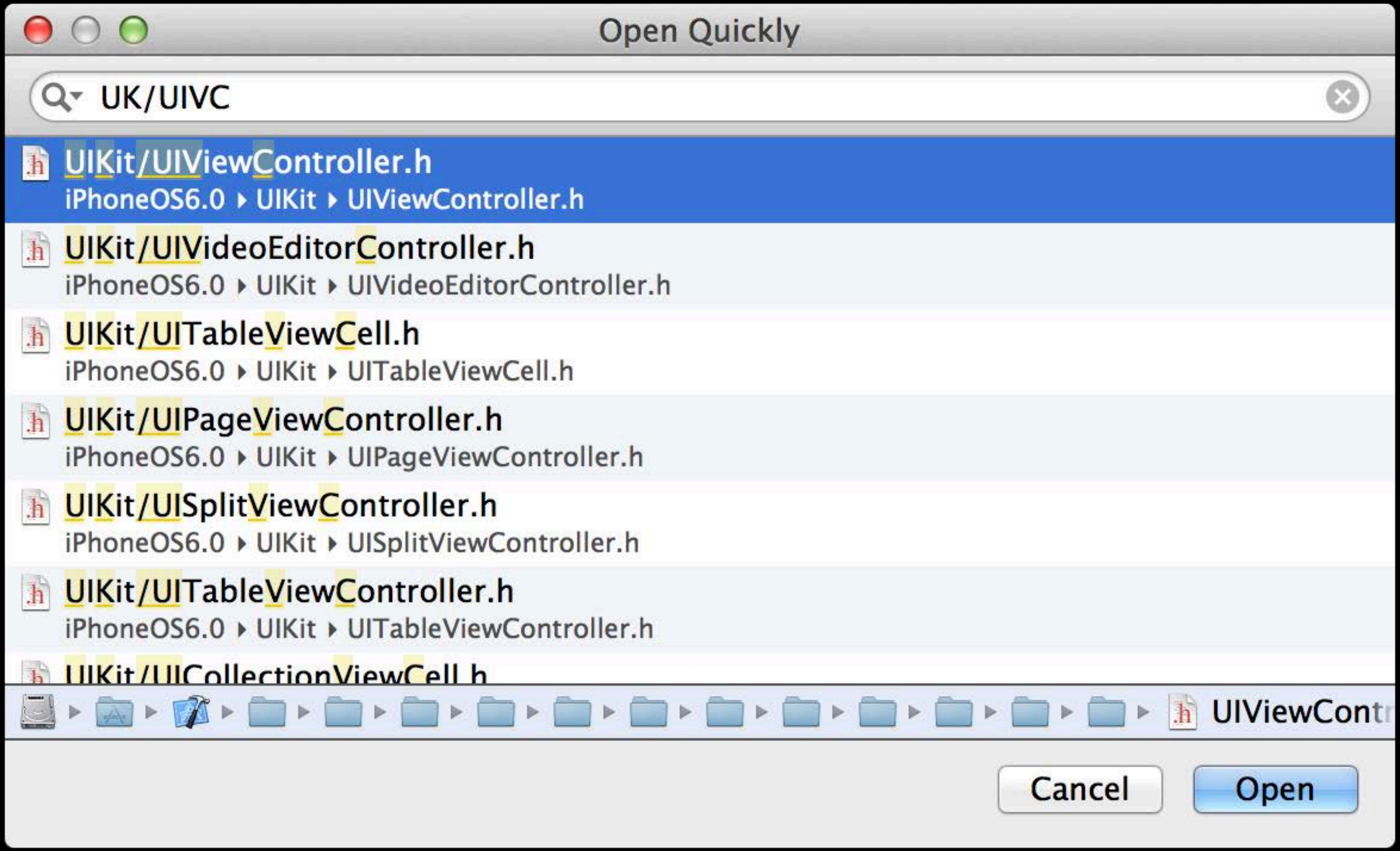
Tip
1


Open Quickly
coretextctpar
CoreText/CTParagraphStyle.h
iPhoneOS6.0 ▸ CoreText ▸ CTParagraphStyle.h
CTParagrap
Cancel
Open

Xcode

Open Quickly (Command-Shift-O)

Tip
1



UK/UIVC

UIKit/UIViewController.h
iPhoneOS6.0 ▸ UIKit ▸ UIViewController.h

UIKit/UIVideoEditorController.h
iPhoneOS6.0 ▸ UIKit ▸ UIVideoEditorController.h

UIKit/UITableViewCell.h
iPhoneOS6.0 ▸ UIKit ▸ UITableViewCell.h

UIKit/UIPageViewController.h
iPhoneOS6.0 ▸ UIKit ▸ UIPageViewController.h

UIKit/UISplitViewController.h
iPhoneOS6.0 ▸ UIKit ▸ UISplitViewController.h

UIKit/UITableViewController.h
iPhoneOS6.0 ▸ UIKit ▸ UITableViewController.h

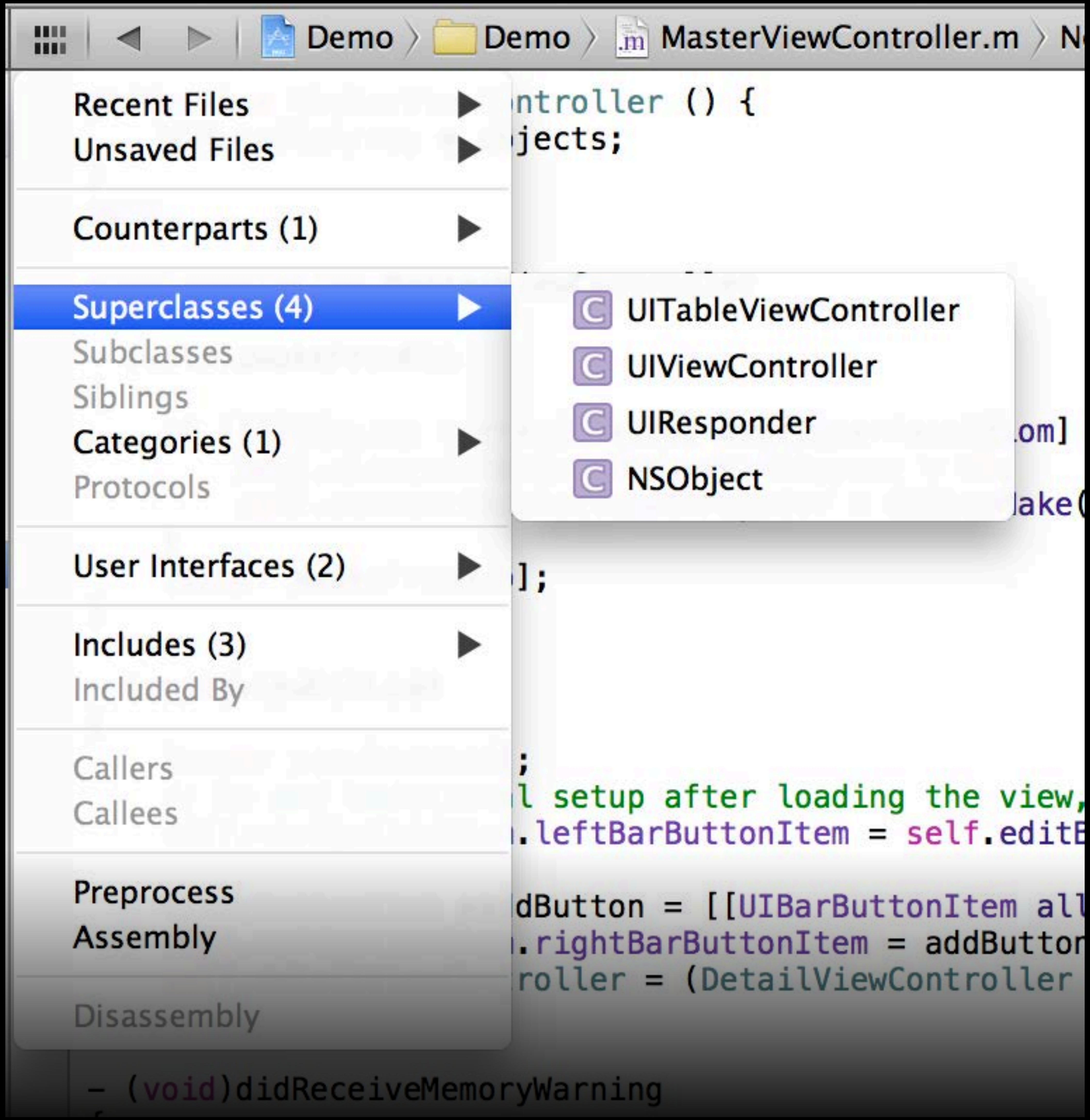
UIKit/UICollectionViewCell.h

UIViewCont

Xcode

Related Files

Tip
2

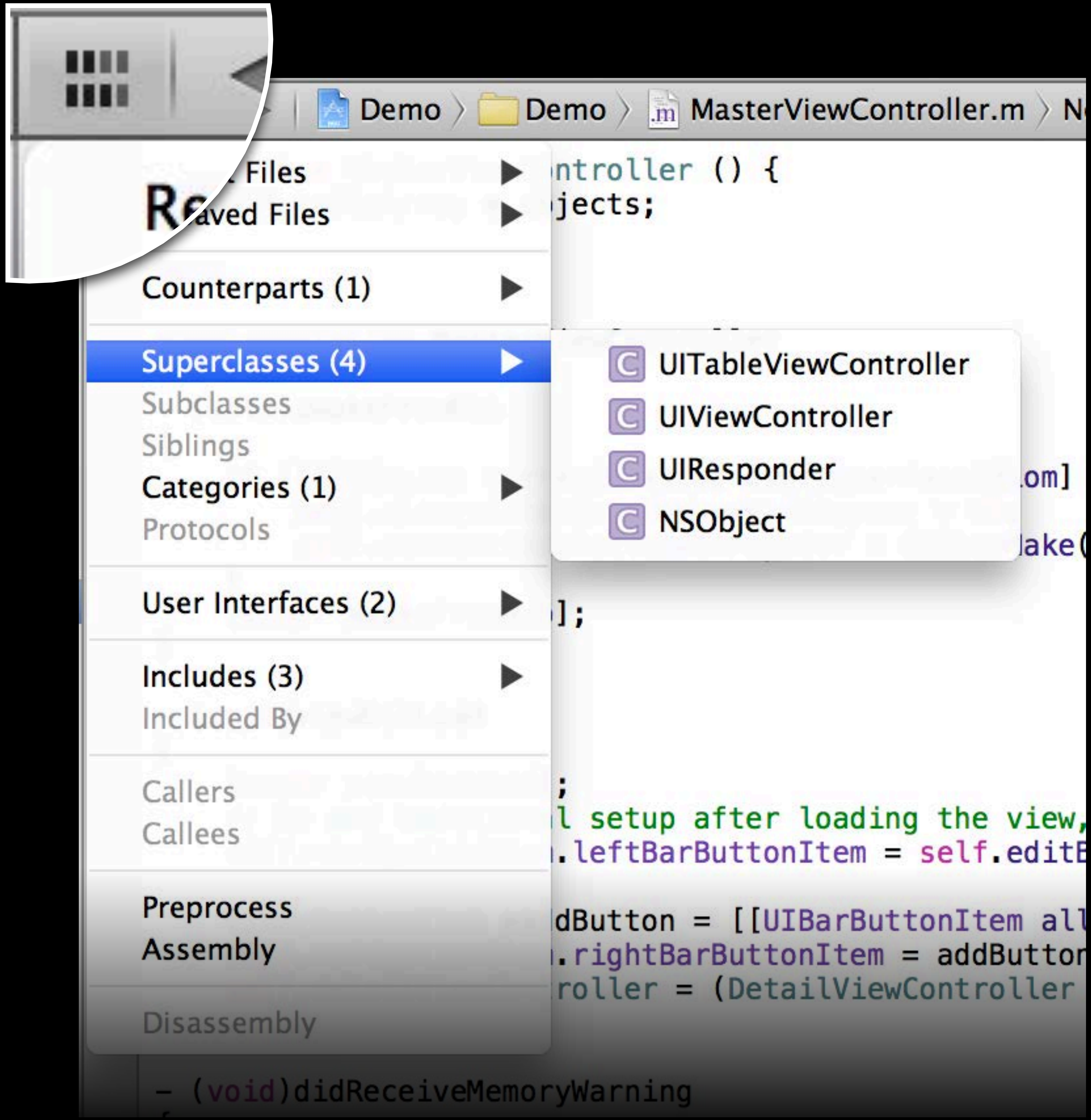


```
ntroller () {
jects;

om]
ake(
];

;
l setup after loading the view,
.leftBarButtonItem = self.editB
dButton = [[UIBarButtonItem all
.rightBarButtonItem = addButton
roller = (DetailViewController
- (void)didReceiveMemoryWarning
```

Xcode

Related Files

Tip
2



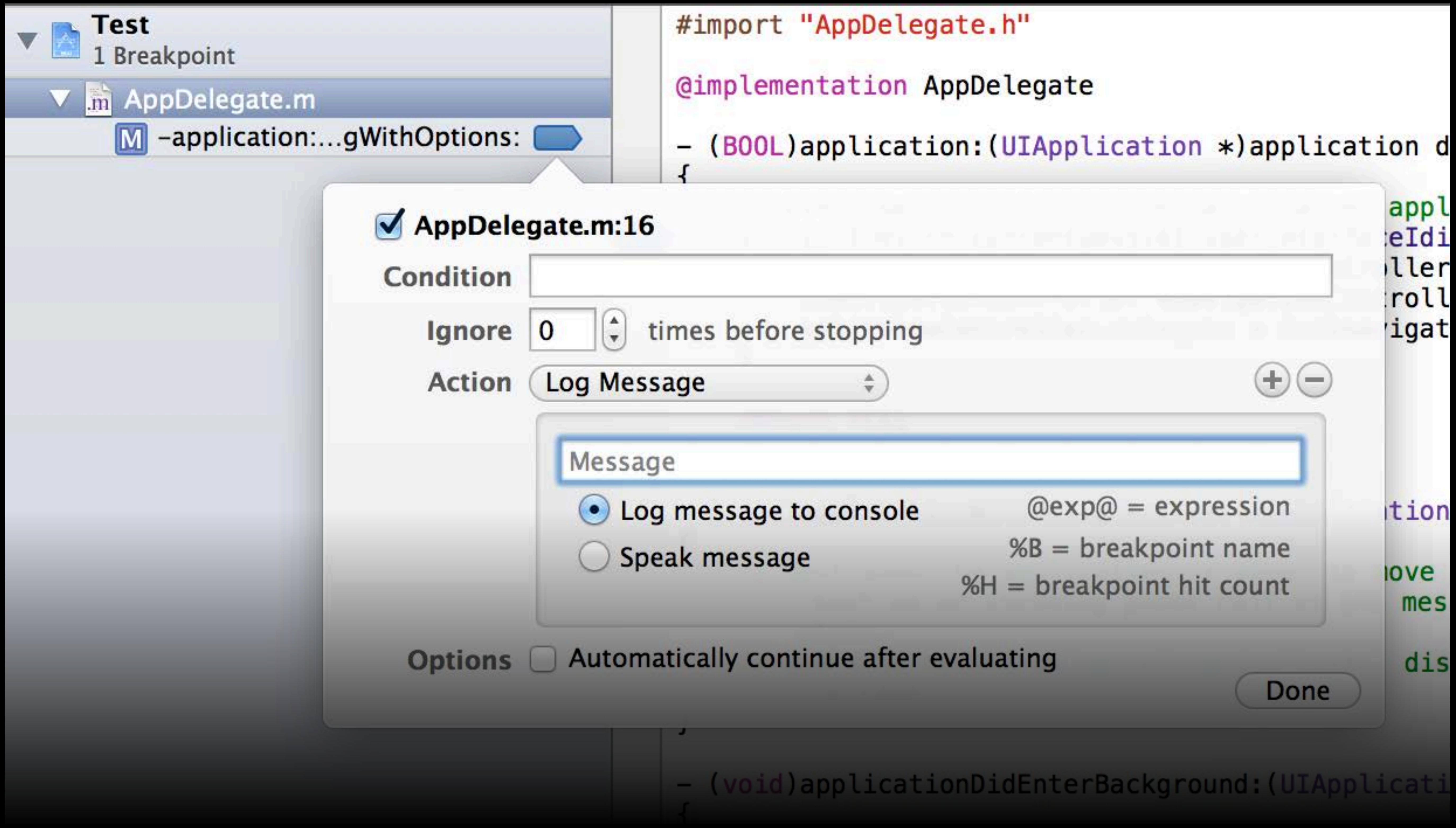
```
ntroller () {
jects;

om]
ake(
];

;
l setup after loading the view,
.leftBarButtonItem = self.editB
dButton = [[UIBarButtonItem all
.rightBarButtonItem = addButton
roller = (DetailViewController
- (void)didReceiveMemoryWarning
```

Xcode

Breakpoint Actions

Tip 3

- Less time recompiling
- Log message, trigger command
- Conditionals



Test
1 Breakpoint
AppDelegate.m
-application:...gWithOptions:

```
#import "AppDelegate.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application d
{
```

AppDelegate.m:16 [✓]
Condition:
Ignore: 0 times before stopping
Action: Log Message
Message
[✓ Log message to console] Speak message
@exp@ = expression
%B = breakpoint name
%H = breakpoint hit count
Options: Automatically continue after evaluating
Done

```
- (void)applicationDidEnterBackground:(UIApplicati
```

Xcode

Debug description

Tip

4

Xcode

Debug description

Tip 4

- Verbose debugging string

Xcode

Debug description

Tip

4

- Verbose debugging string

```
- (NSString *)debugDescription {
```

Xcode

Debug description

Tip

4

- Verbose debugging string

```
- (NSString *)debugDescription {
    return @"A reeeeeeeaaaallly long debug string with nerdy details.";
```

Xcode

Debug description

Tip
4

- Verbose debugging string

```
- (NSString *)debugDescription {
    return @"A reeeeeeeaaaalllly long debug string with nerdy details.";
}
```

Xcode

Debug description

Tip 4

- Verbose debugging string

```
- (NSString *)debugDescription {
    return @"A reeeeeeeaaaalllly long debug string with nerdy details.";
}

(lldb) po self.rootViewController
```

Xcode

Debug description

Tip 4

- Verbose debugging string

```
- (NSString *)debugDescription {
    return @"A reeeeeeeaaaallly long debug string with nerdy details.";
}

(lldb) po self.rootViewController
A reeeeeeeaaaallly long debug string with nerdy details.
```

Xcode

Debug description

Tip 4

- Verbose debugging string

```
- (NSString *)debugDescription {
    return @"A reeeeeeeaaaallly long debug string with nerdy details.";
}

(lldb) po self.rootViewController
A reeeeeeeaaaallly long debug string with nerdy details.
```

- Use with Breakpoint Actions!

Xcode

Recursive description

Tip

5

Xcode

Recursive description

Tip

5

```
<UINavigationItemView: 0x757be70; frame = (350 9; 68 27); opaque = NO;
userInteractionEnabled = NO; layer = <CALayer: 0x757bf60>>
```

Xcode

Recursive description

Tip

5

```
<UINavigationItemView: 0x757be70; frame = (350 9; 68 27); opaque = NO;
userInteractionEnabled = NO; layer = <CALayer: 0x757bf60>>
<UINavigationButton: 0x7586ed0; frame = (7 7; 51 30); opaque = NO; layer =
<CALayer: 0x75870b0>>
```

Xcode

Recursive description

Tip 5

```
<UINavigationItemView: 0x757be70; frame = (350 9; 68 27); opaque = NO;
userInteractionEnabled = NO; layer = <CALayer: 0x757bf60>>
<UINavigationButton: 0x7586ed0; frame = (7 7; 51 30); opaque = NO; layer =
<CALayer: 0x75870b0>>
   | <UIImageView: 0x7588ed0; frame = (0 0; 51 30); opaque = NO;
userInteractionEnabled = NO; layer = <CALayer: 0x7588f30>>
```

Xcode

Recursive description

Tip

5

```
<UINavigationItemView: 0x757be70; frame = (350 9; 68 27); opaque = NO;
userInteractionEnabled = NO; layer = <CALayer: 0x757bf60>>
<UINavigationButton: 0x7586ed0; frame = (7 7; 51 30); opaque = NO; layer =
<CALayer: 0x75870b0>>
   | <UIImageView: 0x7588ed0; frame = (0 0; 51 30); opaque = NO;
userInteractionEnabled = NO; layer = <CALayer: 0x7588f30>>
   | <UIButtonLabel: 0x7587c50; frame = (14 7; 23 15); text = 'Edit';
clipsToBounds = YES; opaque = NO; userInteractionEnabled = NO; layer =
<CALayer: 0x7587d40>>
```

Xcode

Recursive description

Tip 5

```
<UINavigationItemView: 0x757be70; frame = (350 9; 68 27); opaque = NO;
userInteractionEnabled = NO; layer = <CALayer: 0x757bf60>>
<UINavigationButton: 0x7586ed0; frame = (7 7; 51 30); opaque = NO; layer =
<CALayer: 0x75870b0>>
   | <UIImageView: 0x7588ed0; frame = (0 0; 51 30); opaque = NO;
userInteractionEnabled = NO; layer = <CALayer: 0x7588f30>>
   | <UIButtonLabel: 0x7587c50; frame = (14 7; 23 15); text = 'Edit';
clipsToBounds = YES; opaque = NO; userInteractionEnabled = NO; layer =
<CALayer: 0x7587d40>>
```

Objective-C

Language tips

Objective-C
Subscripting

Tip
6

Objective-C
Subscripting

Tip
6

```
NSMutableArray *indexedValues = [NSMutableArray array];
```

Objective-C

Subscripting

Tip
6

```
NSMutableArray *indexedValues = [NSMutableArray array];
indexedValues[0] = @"One";
```

Objective-C

Subscripting

Tip 6

```
NSMutableArray *indexedValues = [NSMutableArray array];
indexedValues[0] = @"One";
NSLog(@"value: %@", indexedValues[0]);
```

Objective-C

Subscripting

Tip

6

```
NSMutableArray *indexedValues = [NSMutableArray array];
indexedValues[0] = @"One";
NSLog(@"value: %@", indexedValues[0]);

NSMutableDictionary *keyedValues = [NSMutableDictionary dictionary];
```

Objective-C
Subscripting

Tip 6

```
NSMutableArray *indexedValues = [NSMutableArray array];
indexedValues[0] = @"One";
NSLog(@"value: %@", indexedValues[0]);

NSMutableDictionary *keyedValues = [NSMutableDictionary dictionary];
keyedValues[@"color"] = [UIColor blueColor];
```

Objective-C
Subscripting

Tip
6

```
NSMutableArray *indexedValues = [NSMutableArray array];
indexedValues[0] = @"One";
NSLog(@"value: %@", indexedValues[0]);

NSMutableDictionary *keyedValues = [NSMutableDictionary dictionary];
keyedValues[@"color"] = [UIColor blueColor];
NSLog(@"value: %@", keyedValues[@"color"]);
```

Objective-C

Custom-indexed subscripting

Tip

6

Objective-C

Custom-indexed subscripting

Tip 6

- Declaration

Objective-C

Custom-indexed subscripting

Tip 6

- Declaration

```
@interface RecordSet : NSObject
```

Objective-C

Custom-indexed subscripting

Tip
6

- Declaration

```
@interface RecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
```

Tip

6

Objective-C

Custom-indexed subscripting

- Declaration

```
@interface RecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
```

Tip

6

Objective-C

Custom-indexed subscripting

- Declaration

```
@interface RecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx;
```

Objective-C

Custom-indexed subscripting

Tip 6

- Declaration

```
@interface RecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx;
@end
```

Tip
6

Objective-C
Custom-indexed subscripting

- Declaration

```
@interface RecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx;
@end
```

- Implementation

```
- (id)objectAtIndexedSubscript:(NSUInteger)idx {
    return [self.indexedValues objectAtIndex:idx];
}

- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx {
    [self.indexedValues insertObject:obj atIndex:idx];
}
```

Objective-C
Custom-indexed subscripting

Tip 6

- Declaration

```
@interface RecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx;
@end
```

- Implementation

```
- (id)objectAtIndexedSubscript:(NSUInteger)idx {
    return [self.indexedValues objectAtIndex:idx];
}

- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx {
    [self.indexedValues insertObject:obj atIndex:idx];
}
```

Objective-C

Custom-indexed subscripting

Tip

6

Objective-C

Custom-indexed subscripting

Tip 6

- In use

Objective-C

Custom-indexed subscripting

Tip 6

- In use

```
RecordSet *recordSet = [[RecordSet alloc] init];
```

Objective-C

Custom-indexed subscripting

- In use

```
RecordSet *recordSet = [[RecordSet alloc] init];
recordSet[0] = [Person personWithName:@"Ana"];
```

Objective-C

Custom-indexed subscripting

Tip 6

- In use

```
RecordSet *recordSet = [[RecordSet alloc] init];
recordSet[0] = [Person personWithName:@"Ana"];
recordSet[1] = [Person personWithName:@"Dave"];
```

Objective-C

Custom-indexed subscripting

Tip 6

- In use

```
RecordSet *recordSet = [[RecordSet alloc] init];
recordSet[0] = [Person personWithName:@"Ana"];
recordSet[1] = [Person personWithName:@"Dave"];
view.person = recordSet[1];
```

Objective-C

Custom-indexed subscripting

Tip 6

- In use

```
RecordSet *recordSet = [[RecordSet alloc] init];
recordSet[0] = [Person personWithName:@"Ana"];
recordSet[1] = [Person personWithName:@"Dave"];
view.person = recordSet[1];
```

Objective-C

Custom-keyed subscripting

Tip

6

Objective-C

Custom-keyed subscripting

Tip 6

- Declaration

Tip

6

Objective-C

Custom-keyed subscripting

- Declaration

```
@interface Person : NSObject
```

Tip
6

Objective-C

Custom-keyed subscripting

- Declaration

```
@interface Person : NSObject
@property (strong) NSMutableDictionary *keyedValues;
```

Objective-C

Custom-keyed subscripting

Tip
6

- Declaration

```
@interface Person : NSObject
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
```

Objective-C

Custom-keyed subscripting

Tip 6

- Declaration

```
@interface Person : NSObject
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key;
```

Tip 6

Objective-C

Custom-keyed subscripting

- Declaration

```
@interface Person : NSObject
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key;
@end
```

Objective-C

Custom-keyed subscripting

Tip 6

- Declaration

```
@interface Person : NSObject
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key;
@end
```

Objective-C

Custom-keyed subscripting

Tip 6

- Declaration

```
@interface Person : NSObject
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key;
@end
```

- Implementation

```
- (id)objectAtKeyedSubscript:(id <NSCopying>)key {
    return [self.keyedValues objectForKey:key];
}

- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key {
    [self.keyedValues setObject:obj forKey:key];
}
```

Objective-C

Custom-keyed subscripting

Tip
6

- Declaration

```
@interface Person : NSObject
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key;
@end
```

- Implementation

```
- (id)objectAtKeyedSubscript:(id <NSCopying>)key {
    return [self.keyedValues objectForKey:key];
}

- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key {
    [self.keyedValues setObject:obj forKey:key];
}
```

Objective-C

Custom-keyed subscripting

Tip 6

- Declaration

```
@interface Person : NSObject
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key;
@end
```

- Implementation

```
- (id)objectAtKeyedSubscript:(id <NSCopying>)key {
    return [self.keyedValues objectForKey:key];
}

- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key {
    [self.keyedValues setObject:obj forKey:key];
}
```

Objective-C

Custom-keyed subscripting

Tip

6

Objective-C

Custom-keyed subscripting

Tip
6

- In use

Objective-C

Custom-keyed subscripting

Tip 6

- In use

```
Person *person = [[Person alloc] init];
```

Objective-C
Custom-keyed subscripting

Tip
6

- In use

```
Person *person = [[Person alloc] init];
person[@"favoriteColor"] = [UIColor blueColor];
```

Objective-C
Custom-keyed subscripting

Tip
6

- In use

```
Person *person = [[Person alloc] init];
person[@"favoriteColor"] = [UIColor blueColor];
view.backgroundColor = person[@"favoriteColor"];
```

Objective-C

Custom-keyed subscripting

Tip 6

- In use

```
Person *person = [[Person alloc] init];
person[@"favoriteColor"] = [UIColor blueColor];
view.backgroundColor = person[@"favoriteColor"];
```

Objective-C

Combined subscripting

Tip

6

Objective-C

Combined subscripting

Tip 6

- Declaration

Objective-C

Combined subscripting

Tip
6

- Declaration

```
@interface UltraRecordSet : NSObject
```

Objective-C

Combined subscripting

Tip
6

- Declaration

```
@interface UltraRecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
```

Objective-C

Combined subscripting

Tip

6

- Declaration

```
@interface UltraRecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
@property (strong) NSMutableDictionary *keyedValues;
```

Tip 6

Objective-C

Combined subscripting

- Declaration

```
@interface UltraRecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
```

Objective-C

Combined subscripting

Tip 6

- Declaration

```
@interface UltraRecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx;
```

Objective-C

Combined subscripting

Tip
6

- Declaration

```
@interface UltraRecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
```

Objective-C

Combined subscripting

Tip 6

- Declaration

```
@interface UltraRecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key;
```

Objective-C

Combined subscripting

Tip 6

- Declaration

```
@interface UltraRecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key;
@end
```

Objective-C

Combined subscripting

Tip 6

- Declaration

```
@interface UltraRecordSet : NSObject
@property (strong) NSMutableArray *indexedValues;
@property (strong) NSMutableDictionary *keyedValues;
- (id)objectAtIndexedSubscript:(NSUInteger)idx;
- (void)setObject:(id)obj atIndexedSubscript:(NSUInteger)idx;
- (id)objectAtKeyedSubscript:(id <NSCopying>)key;
- (void)setObject:(id)obj forKeyedSubscript:(id <NSCopying>)key;
@end
```

Objective-C

Combined subscripting

Tip

6

Objective-C

Combined subscripting

Tip 6

- In use

Objective-C

Combined subscripting

Tip 6

- In use

```
UltraRecordSet *recordSet = [[UltraRecordSet alloc] init];
```

Objective-C

Combined subscripting

Tip 6

- In use

```
UltraRecordSet *recordSet = [[UltraRecordSet alloc] init];
recordSet[@"cityName"] = @"Cupertino";
```

Objective-C

Combined subscripting

Tip

6

- In use

```
UltraRecordSet *recordSet = [[UltraRecordSet alloc] init];
recordSet[@"cityName"] = @"Cupertino";
recordSet[0] = [Person personWithName:@"Ana"];
```

Objective-C

Combined subscripting

Tip 6

- In use

```
UltraRecordSet *recordSet = [[UltraRecordSet alloc] init];
recordSet[@"cityName"] = @"Cupertino";
recordSet[0] = [Person personWithName:@"Ana"];
recordSet[1] = [Person personWithName:@"Dave"];
```

Objective-C

Combined subscripting

Tip

6

- In use

```
UltraRecordSet *recordSet = [[UltraRecordSet alloc] init];
recordSet[@"cityName"] = @"Cupertino";
recordSet[0] = [Person personWithName:@"Ana"];
recordSet[1] = [Person personWithName:@"Dave"];
view.cityName = recordSet[@"cityName"];
```

Objective-C

Combined subscripting

Tip
6

- In use

```
UltraRecordSet *recordSet = [[UltraRecordSet alloc] init];
recordSet[@"cityName"] = @"Cupertino";
recordSet[0] = [Person personWithName:@"Ana"];
recordSet[1] = [Person personWithName:@"Dave"];
view.cityName = recordSet[@"cityName"];
view.person = recordSet[1];
```

Objective-C
Combined subscripting

Tip 6

- In use

```
UltraRecordSet *recordSet = [[UltraRecordSet alloc] init];
recordSet[@"cityName"] = @"Cupertino";
recordSet[0] = [Person personWithName:@"Ana"];
recordSet[1] = [Person personWithName:@"Dave"];
view.cityName = recordSet[@"cityName"];
view.person = recordSet[1];
```

Objective-C

Reduce code noise

Tip 7

- Private declarations not necessary since Xcode 4.3

```
 @interface Example (PrivateMethods)
 - (NSString *)privateStatusStringForKey:(NSString *)key;
 @end


@implementation Example
- (NSString *)currentStatusString {
     return [self privateStatusStringForKey:self.currentKey];
}

- (NSString *)privateStatusStringForKey:(NSString *)key {
     self.statusStringsByKey[key];
}
@end
```

Objective-C

Reduce code noise

Tip 7

- Private declarations not necessary since Xcode 4.3

```
@interface Example (PrivateMethods)
- (NSString *)privateStatusStringForKey:(NSString *)key;
@end
```

```
@implementation Example
- (NSString *)currentStatusString {
    return [self privateStatusStringForKey:self.currentKey];
}

- (NSString *)privateStatusStringForKey:(NSString *)key {
    self.statusStringsByKey[key];
}
@end
```

Objective-C

Reduce code noise

Tip

7

```
@implementation Example
- (NSString *)currentStatusString {
    return [self privateStatusStringForKey:self.currentKey];
}

- (NSString *)privateStatusStringForKey:(NSString *)key {
    self.statusStringsByKey[key];
}
@end
```

Objective-C

Reduce code noise

Tip

7

Objective-C

Reduce code noise

Tip

7

- Synthesize not necessary for @property since Xcode 4.4

```
@implementation Example
@synthesize currentKey;
@synthesize statusStringsByKey;

- (void)methodName {
   ...
}

@end
```

Objective-C
Reduce code noise

Tip
7

- Synthesize not necessary for @property since Xcode 4.4

```
@implementation Example
@synthesize currentKey;
@synthesize statusStringsByKey;

- (void)methodName {
   ...
}

@end
```

Objective-C

Reduce code noise

Tip 7

- Synthesize not necessary for @property since Xcode 4.4

```
@implementation Example
- (void)methodName {
   ...
}

@end
```

Demo

Xcode

Mattt Thompson
NSHipster.com

Foundation

Data model tips

Scott Stevenson
Software Engineer

Foundation

NSOperation

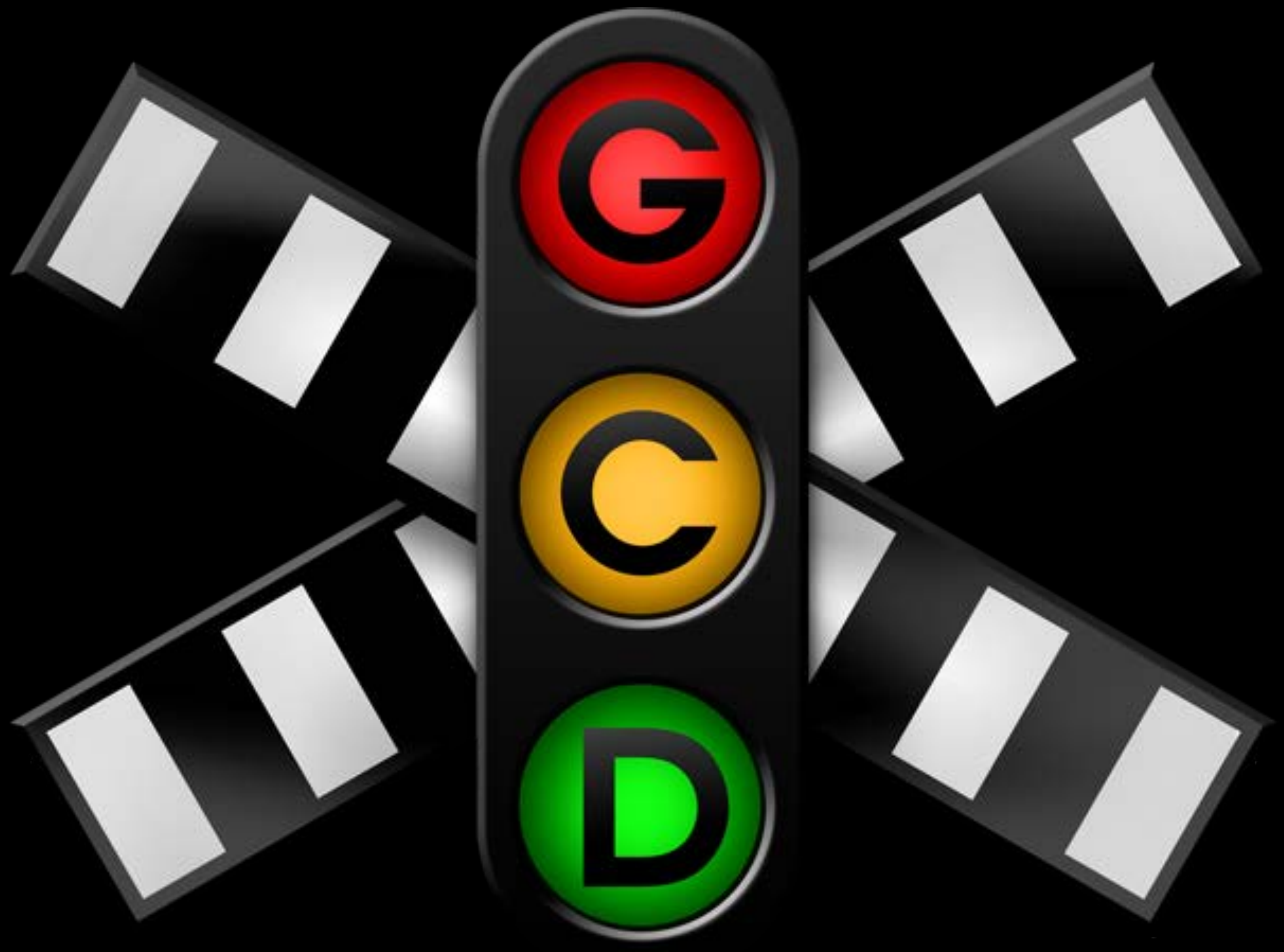
Tip

8

Foundation
NSOperation

Tip 8

- Built on Grand Central Dispatch



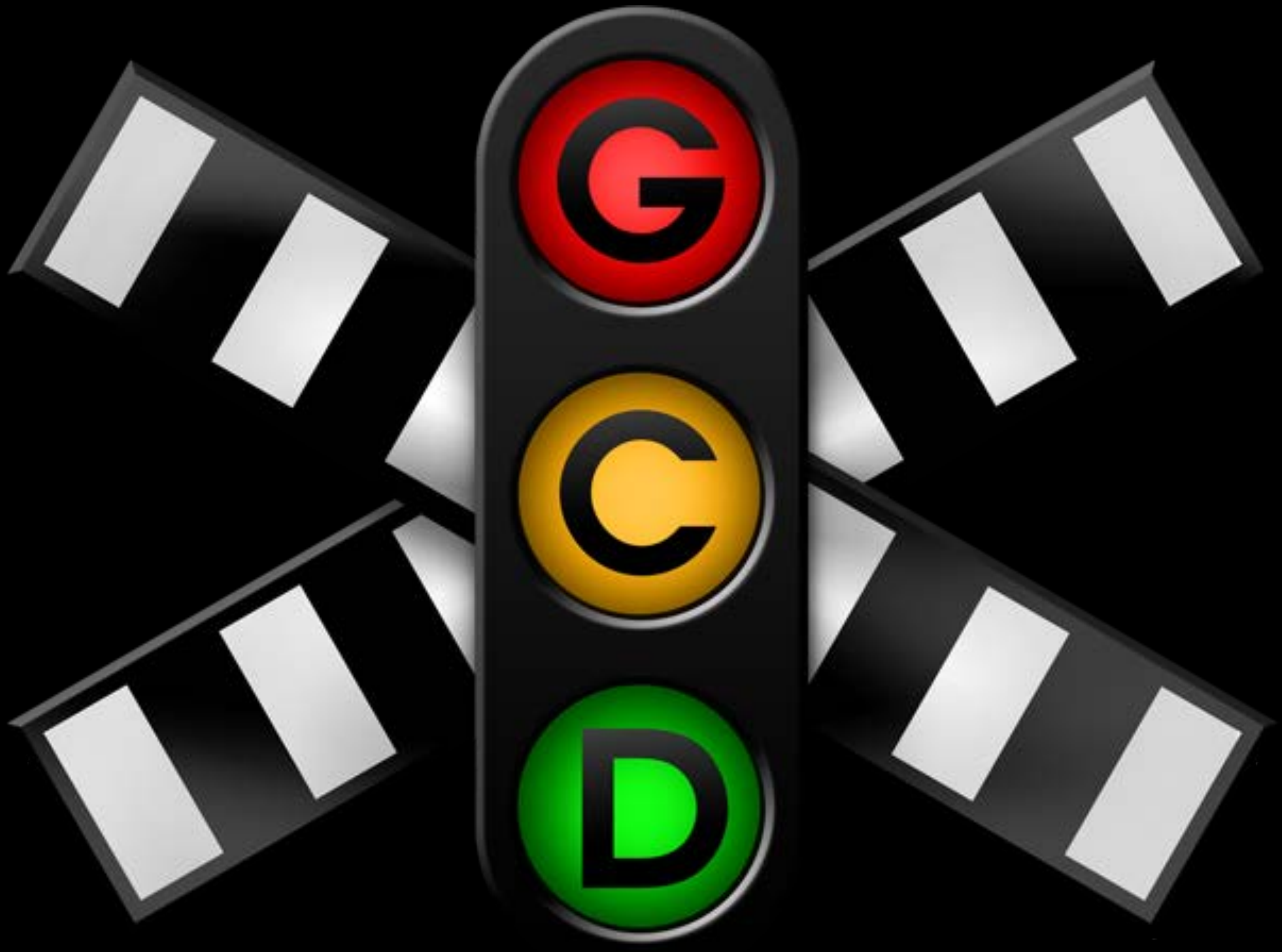
Foundation

NSOperation

Tip

8

- Built on Grand Central Dispatch
- Features

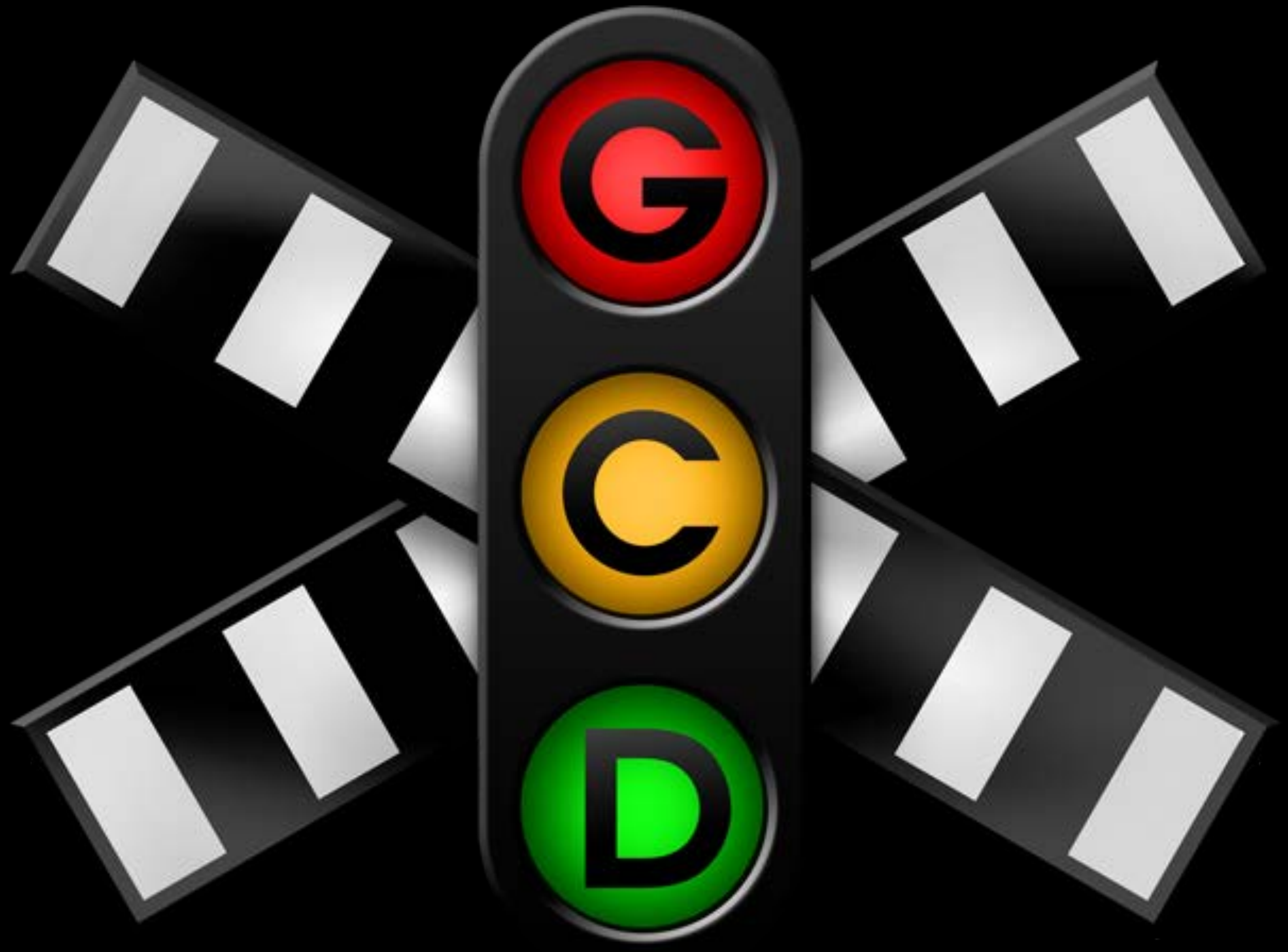


Tip 8

Foundation

NSOperation

- Built on Grand Central Dispatch
- Features
  - Cancellation



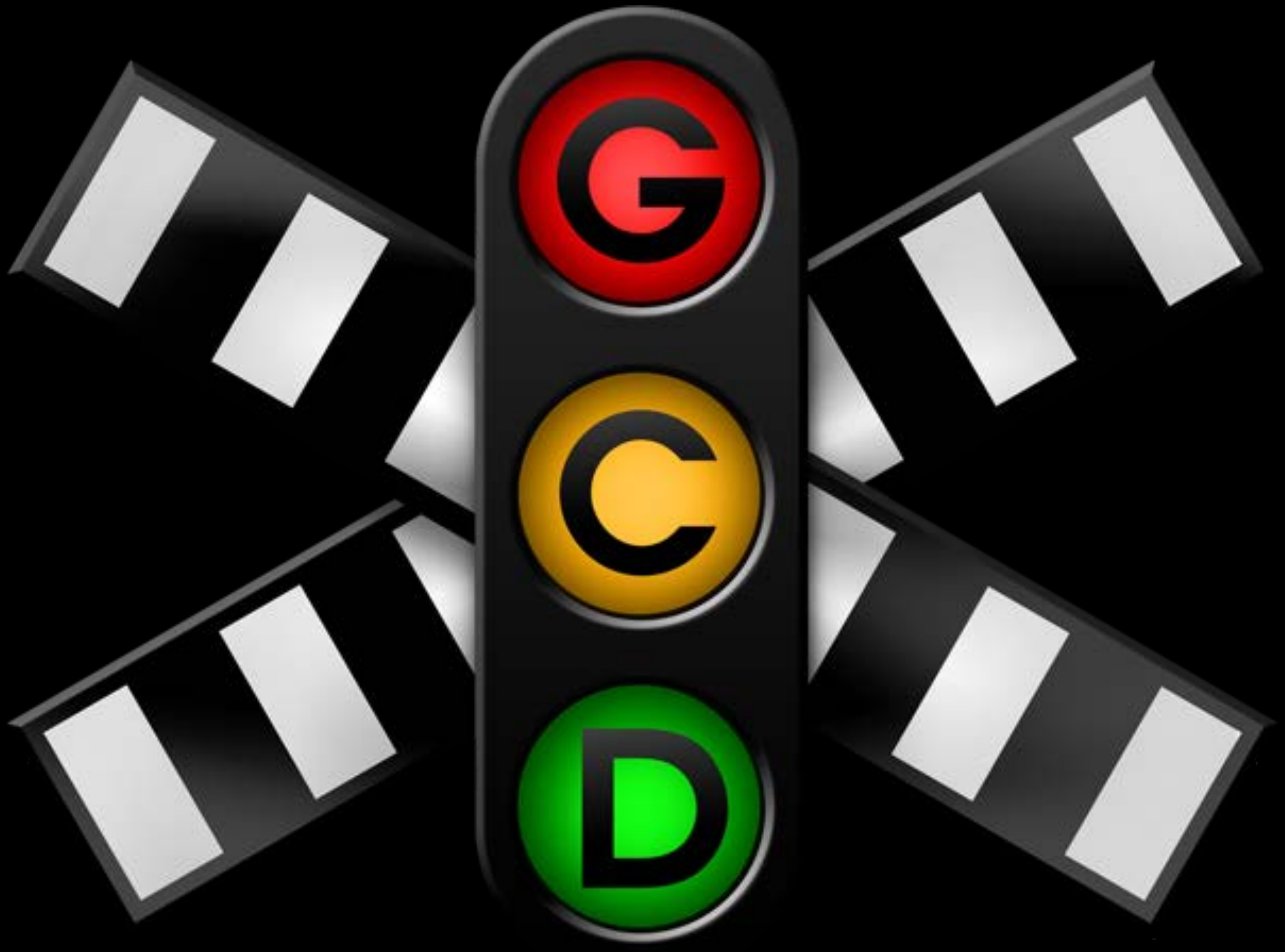
Foundation

NSOperation

Tip

8

- Built on Grand Central Dispatch
- Features
  - Cancellation
  - Max count

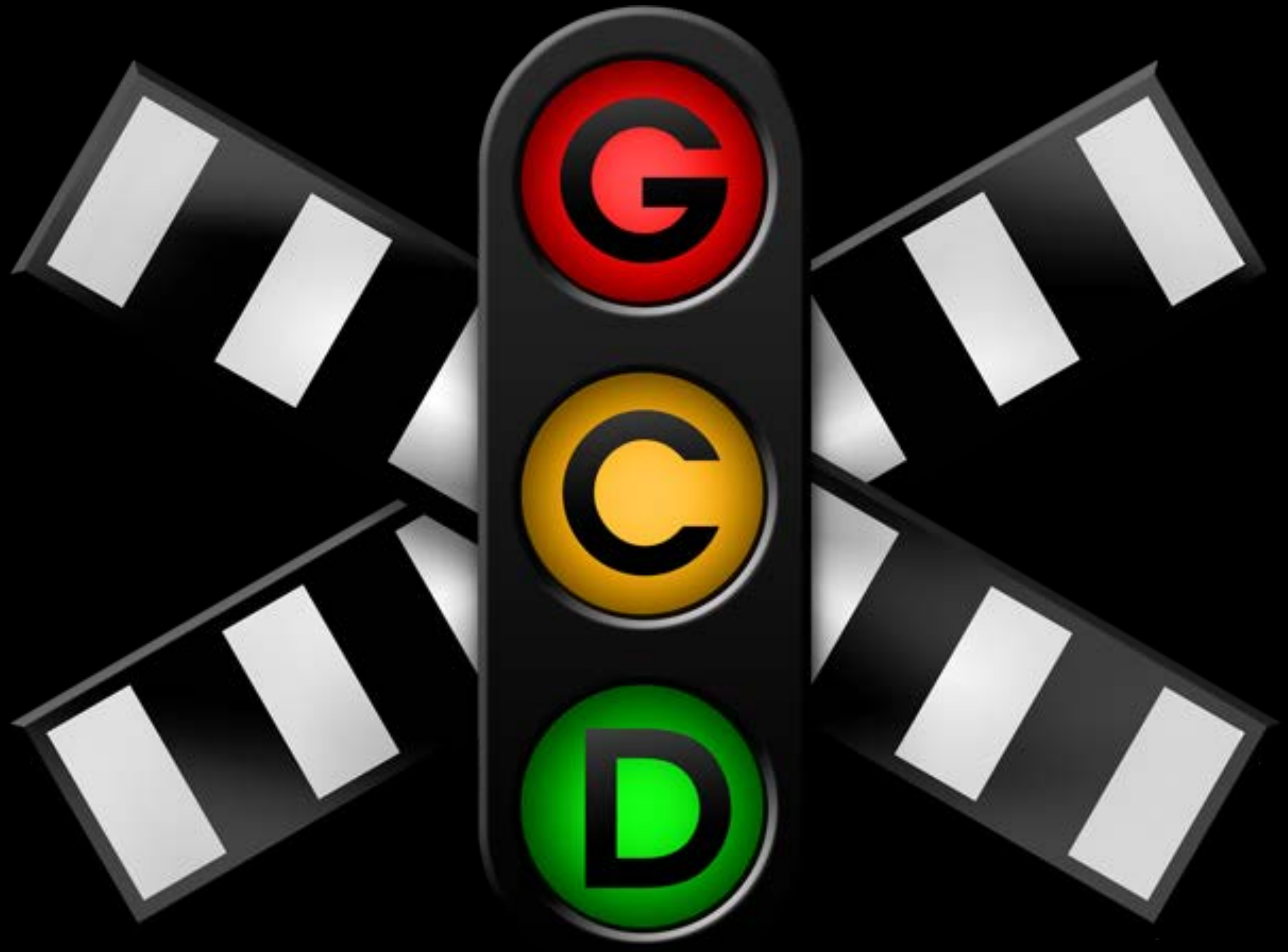


Foundation

NSOperation

Tip 8

- Built on Grand Central Dispatch
- Features
  - Cancellation
  - Max count
  - Dependencies

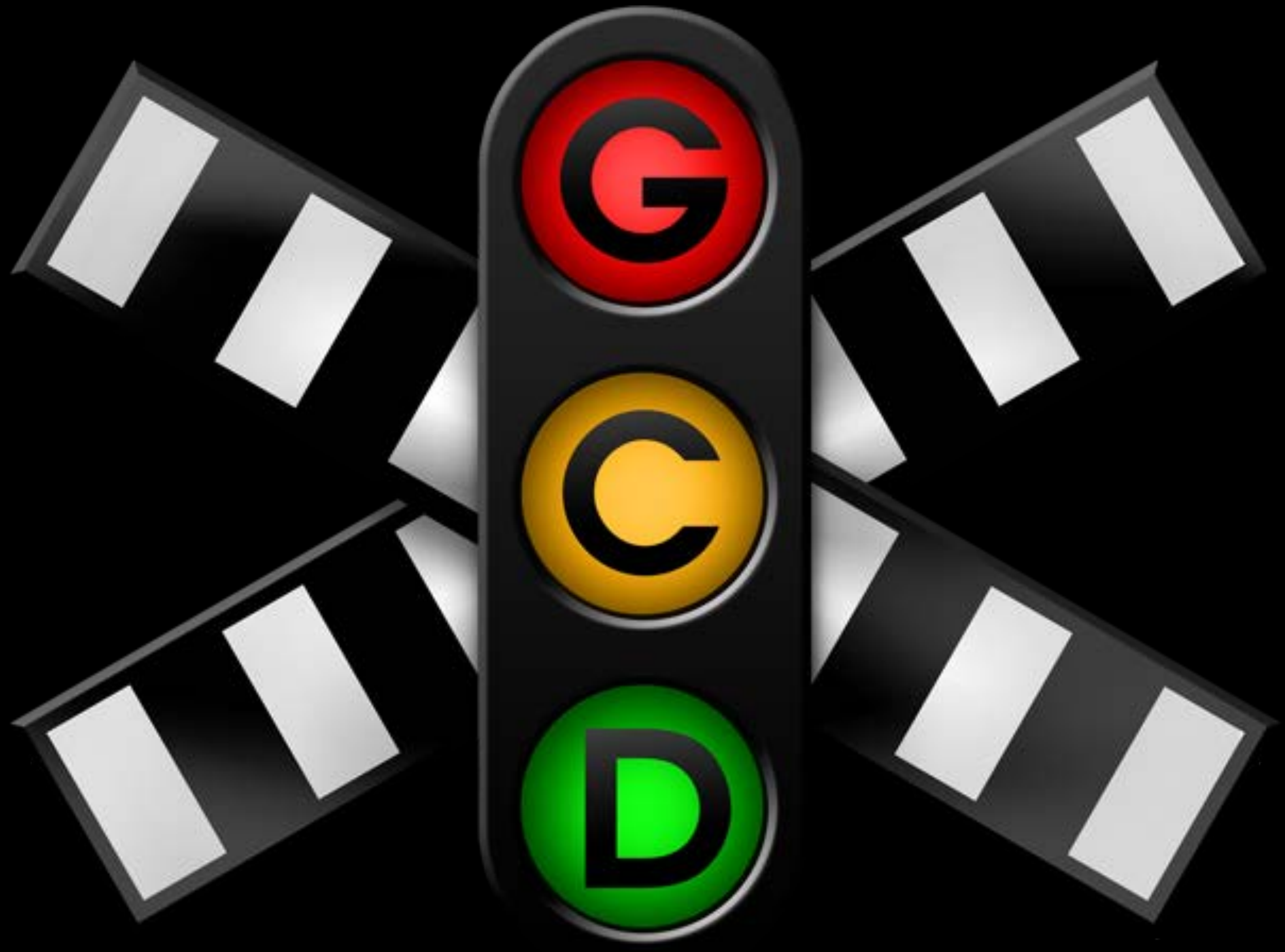


Foundation

NSOperation

Tip 8

- Built on Grand Central Dispatch
- Features
  - Cancellation
  - Max count
  - Dependencies
- Objective-C API

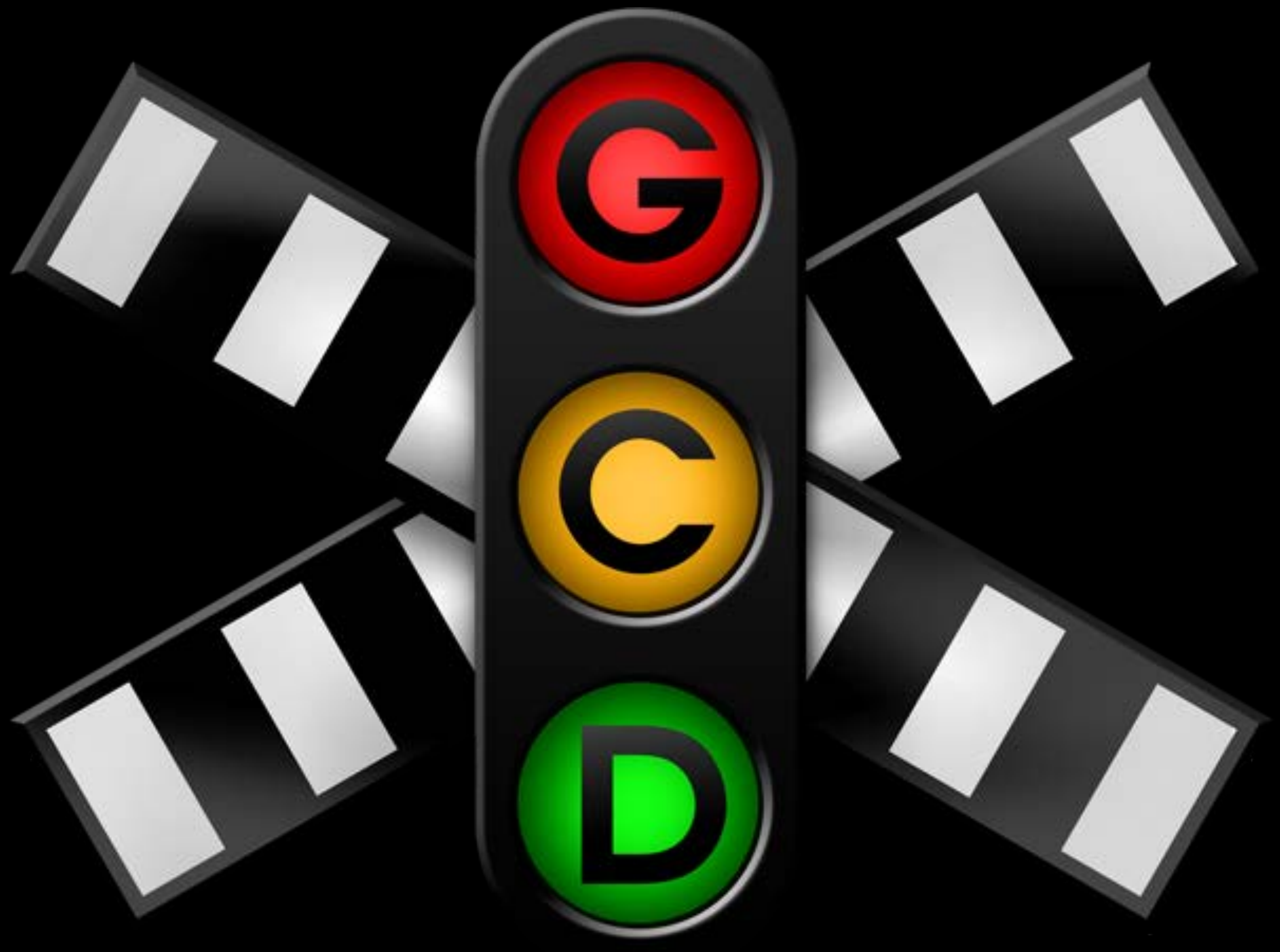


Foundation

NSOperation

Tip 8

- Built on Grand Central Dispatch
- Features
  - Cancellation
  - Max count
  - Dependencies
- Objective-C API
  - Subclasses

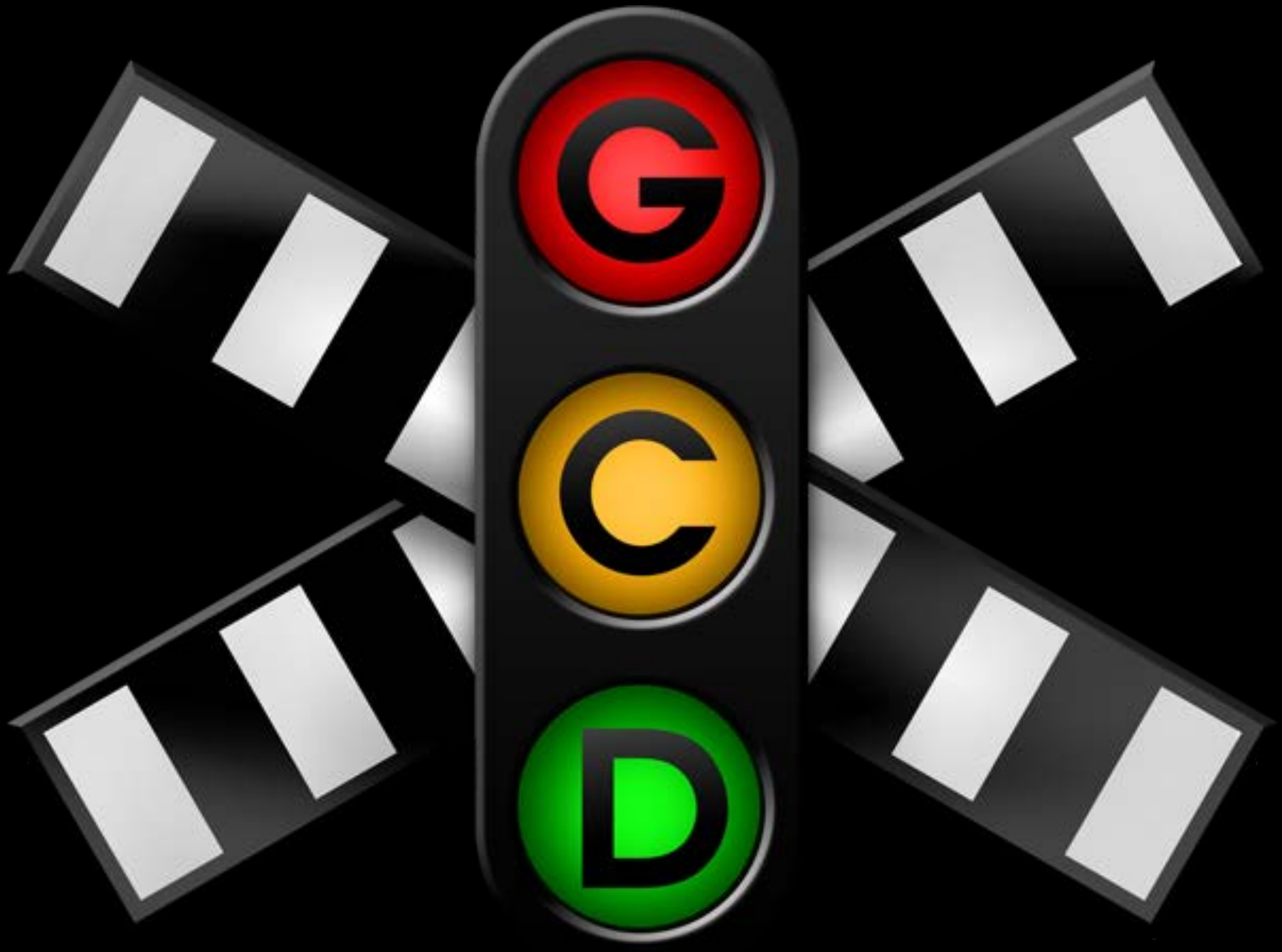


Foundation

NSOperation

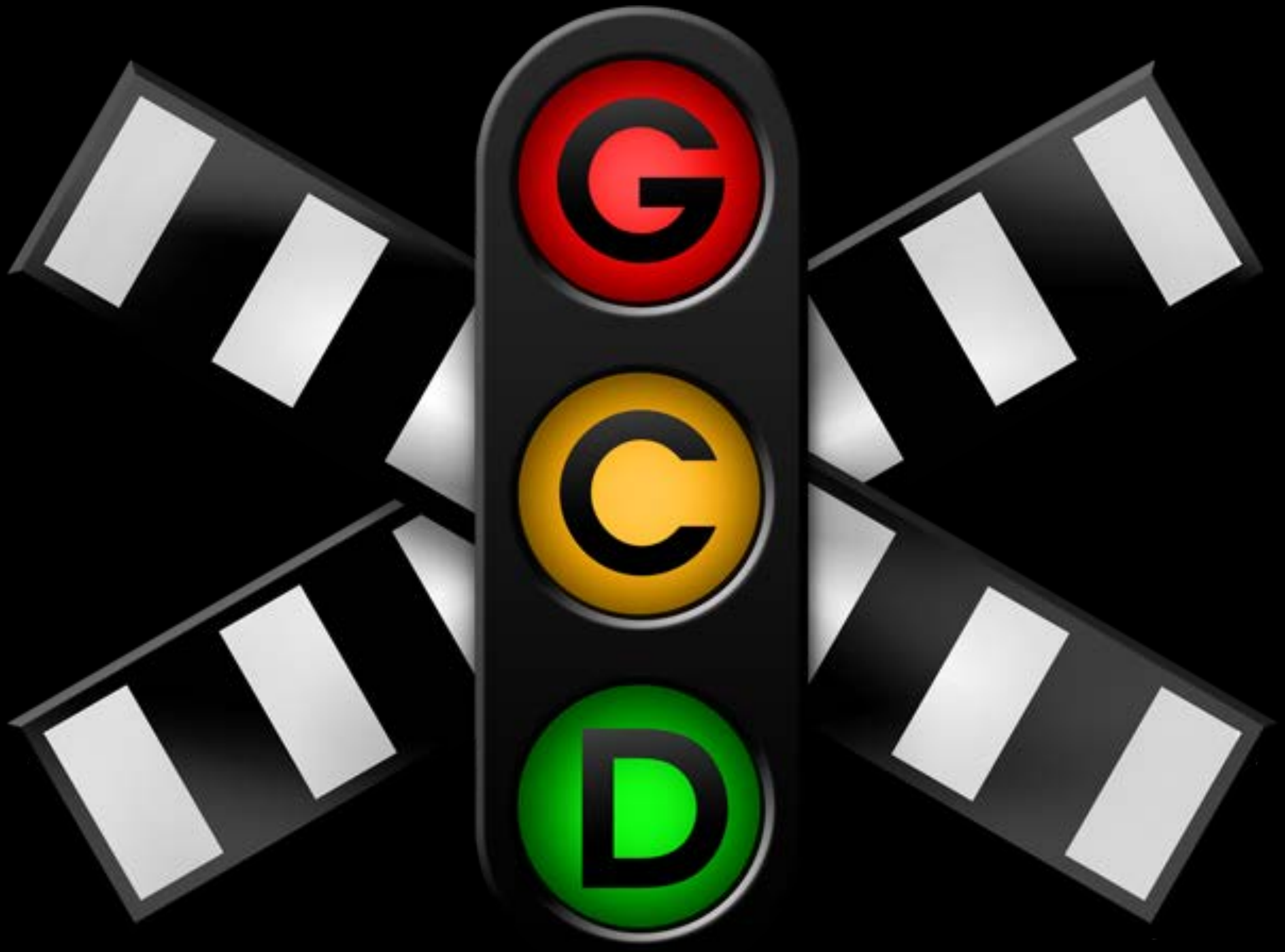
Tip 8

- Built on Grand Central Dispatch
- Features
  - Cancellation
  - Max count
  - Dependencies
- Objective-C API
  - Subclasses
  - Categories



Foundation
NSOperation

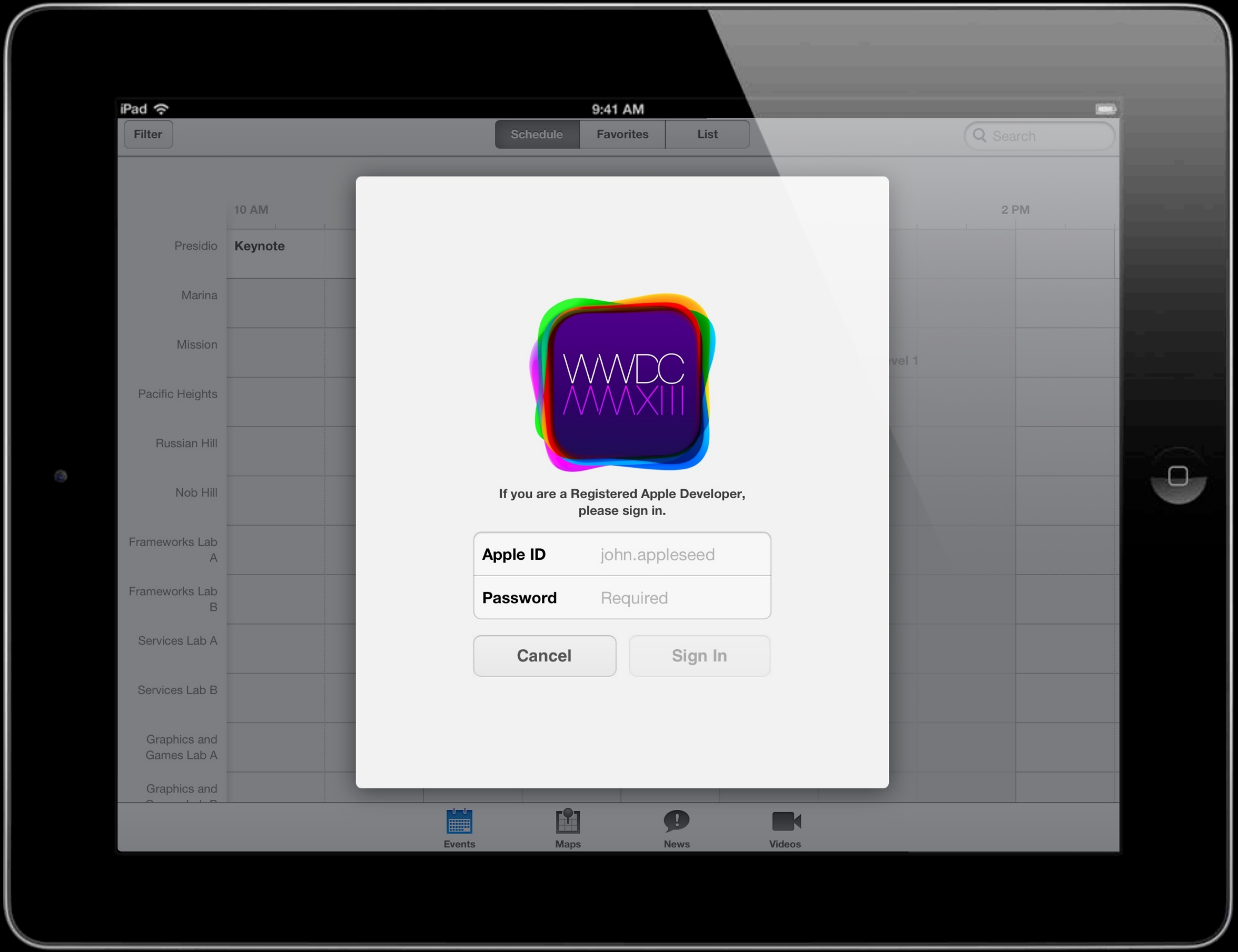
- Built on Grand Central Dispatch
- Features
  - Cancellation
  - Max count
  - Dependencies
- Objective-C API
  - Subclasses
  - Categories
  - Key-Value Observing



Foundation

NSOperation

Tip 8


iPad 9:41 AM

Filter | Schedule | Favorites | List | Search

10 AM 2 PM

Presidio Keynote

Marina

Mission

Pacific Heights

Russian Hill

Nob Hill

Frameworks Lab A

Frameworks Lab B

Services Lab A

Services Lab B

Graphics and Games Lab A

Graphics and

If you are a Registered Apple Developer, please sign in.

Apple ID	john.appleseed
Password	Required

Cancel | Sign In

Foundation

NSOperation dependencies

Tip
8

Foundation

NSOperation dependencies

Tip

8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {
```

Tip
8

Foundation

NSOperation dependencies

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

    self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
```

Foundation

NSOperation dependencies

Tip

8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

    self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
    [operation addDependency:self.authOperation];
```

Foundation

NSOperation dependencies

Tip
8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

    self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
    [operation addDependency:self.authOperation];
    [self.queue addOperation: self.authOperation waitUntilFinished:NO];
```

Foundation

NSOperation dependencies

Tip

8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

     self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
     [operation addDependency:self.authOperation];
     [self.queue addOperation: self.authOperation waitUntilFinished:NO];
}
```

Foundation

NSOperation dependencies

Tip

8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

    self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
    [operation addDependency:self.authOperation];
    [self.queue addOperation: self.authOperation waitUntilFinished:NO];
}

- (void)setFavorite:(BOOL)status forSessionID:(NSString *)sessionID
completion:(void(^)(BOOL))block {
```

Foundation

NSOperation dependencies

Tip

8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

    self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
    [operation addDependency:self.authOperation];
    [self.queue addOperation: self.authOperation waitUntilFinished:NO];
}

- (void)setFavorite:(BOOL)status forSessionID:(NSString *)sessionID
completion:(void(^)(BOOL))block {

    WWDCSetFavoriteOperation *operation;
```

Foundation

NSOperation dependencies

Tip

8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

    self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
    [operation addDependency:self.authOperation];
    [self.queue addOperation: self.authOperation waitUntilFinished:NO];
}

- (void)setFavorite:(BOOL)status forSessionID:(NSString *)sessionID
completion:(void(^)(BOOL))block {

    WWDCSetFavoriteOperation *operation;
    operation =[[WWDCSetFavoriteOperation alloc] initWithSuccessBlock:block];
```

Foundation

NSOperation dependencies

Tip

8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

    self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
    [operation addDependency:self.authOperation];
    [self.queue addOperation: self.authOperation waitUntilFinished:NO];
}

- (void)setFavorite:(BOOL)status forSessionID:(NSString *)sessionID
completion:(void(^)(BOOL))block {

    WWDCSetFavoriteOperation *operation;
    operation =[[WWDCSetFavoriteOperation alloc] initWithSuccessBlock:block];
    operation.favorite = status;
```

Foundation

NSOperation dependencies

Tip 8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

     self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
     [operation addDependency:self.authOperation];
     [self.queue addOperation: self.authOperation waitUntilFinished:NO];
}

 - (void)setFavorite:(BOOL)status forSessionID:(NSString *)sessionID
 completion:(void(^)(BOOL))block {

     WWDCSetFavoriteOperation *operation;
     operation =[[WWDCSetFavoriteOperation alloc] initWithSuccessBlock:block];
     operation.favorite = status;
     [self addDependenciesForAuthorizedOperation:operation];
```

Foundation

NSOperation dependencies

Tip 8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

     self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
     [operation addDependency:self.authOperation];
     [self.queue addOperation: self.authOperation waitUntilFinished:NO];
}

 - (void)setFavorite:(BOOL)status forSessionID:(NSString *)sessionID
 completion:(void(^)(BOOL))block {

     WWDCSetFavoriteOperation *operation;
     operation =[[WWDCSetFavoriteOperation alloc] initWithSuccessBlock:block];
     operation.favorite = status;
     [self addDependenciesForAuthorizedOperation:operation];
     [self.operationQueue addOperation:operation];
```

Foundation

NSOperation dependencies

Tip

8

```
- (void)addDependenciesForAuthorizedOperation:(WWDCOperation *)operation {

     self.authOperation = [[WWDCUserAuthorizationOperation alloc] init];
     [operation addDependency:self.authOperation];
     [self.queue addOperation: self.authOperation waitUntilFinished:NO];
}

- (void)setFavorite:(BOOL)status forSessionID:(NSString *)sessionID
completion:(void(^)(BOOL))block {

     WWDCSetFavoriteOperation *operation;
     operation =[[WWDCSetFavoriteOperation alloc] initWithSuccessBlock:block];
     operation.favorite = status;
     [self addDependenciesForAuthorizedOperation:operation];
     [self.operationQueue addOperation:operation];
}
```

Foundation

NSExpression

Tip

9

Tip 9

Foundation

NSExpression

- Stand back, I'm going to try math

Foundation

NSExpression

Tip

9

- Stand back, I'm going to try math

```
NSString *text = @"3 + 5 * 4e10";
```

Foundation
NSExpression

Tip
9

- Stand back, I'm going to try math

```
NSString *text = @"3 + 5 * 4e10";
NSExpression *e = [NSExpression expressionWithFormat:text, nil];
```

Foundation
NSExpression

Tip 9

- Stand back, I'm going to try math

```
NSString *text = @"3 + 5 * 4e10";
NSExpression *e = [NSExpression expressionWithFormat:text, nil];
NSNumber *result = [e expressionValueWithObject:nil context:nil];
```

Foundation

NSExpression

Tip 9

- Stand back, I'm going to try math

```
NSString *text = @"3 + 5 * 4e10";
NSExpression *e = [NSExpression expressionWithFormat:text, nil];
NSNumber *result = [e expressionValueWithObject:nil context:nil];
NSLog(@"result: %@", result);
```

Foundation

NSExpression

Tip
9

- Stand back, I'm going to try math

```
NSString *text = @"3 + 5 * 4e10";
NSExpression *e = [NSExpression expressionWithFormat:text, nil];
NSNumber *result = [e expressionValueWithObject:nil context:nil];
NSLog(@"result: %@", result);
```

```
result: 200000000003
```

Foundation

NSSet and NSOrderedSet

Tip

10

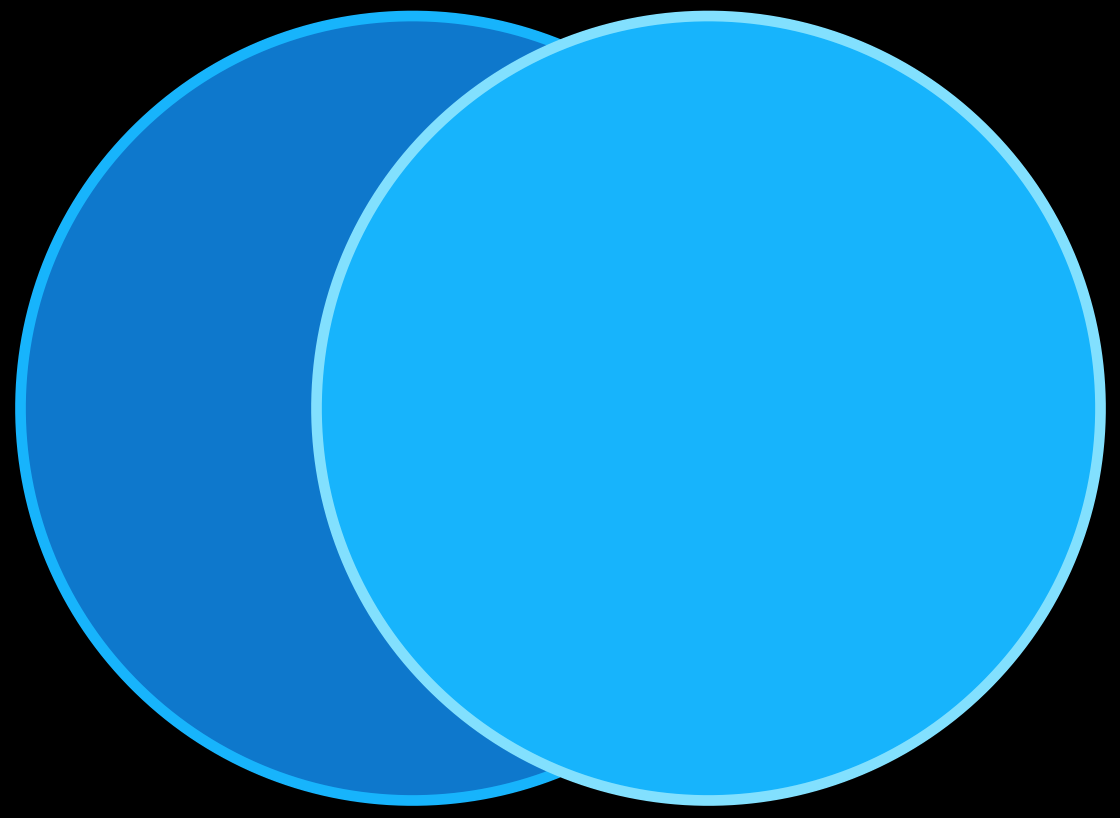
- Guaranteed uniqueness, fast membership lookup
- Membership cache
- Set calculations

Foundation

NSSet and NSOrderedSet

Tip 10

- Guaranteed uniqueness, fast membership lookup
- Membership cache
- Set calculations



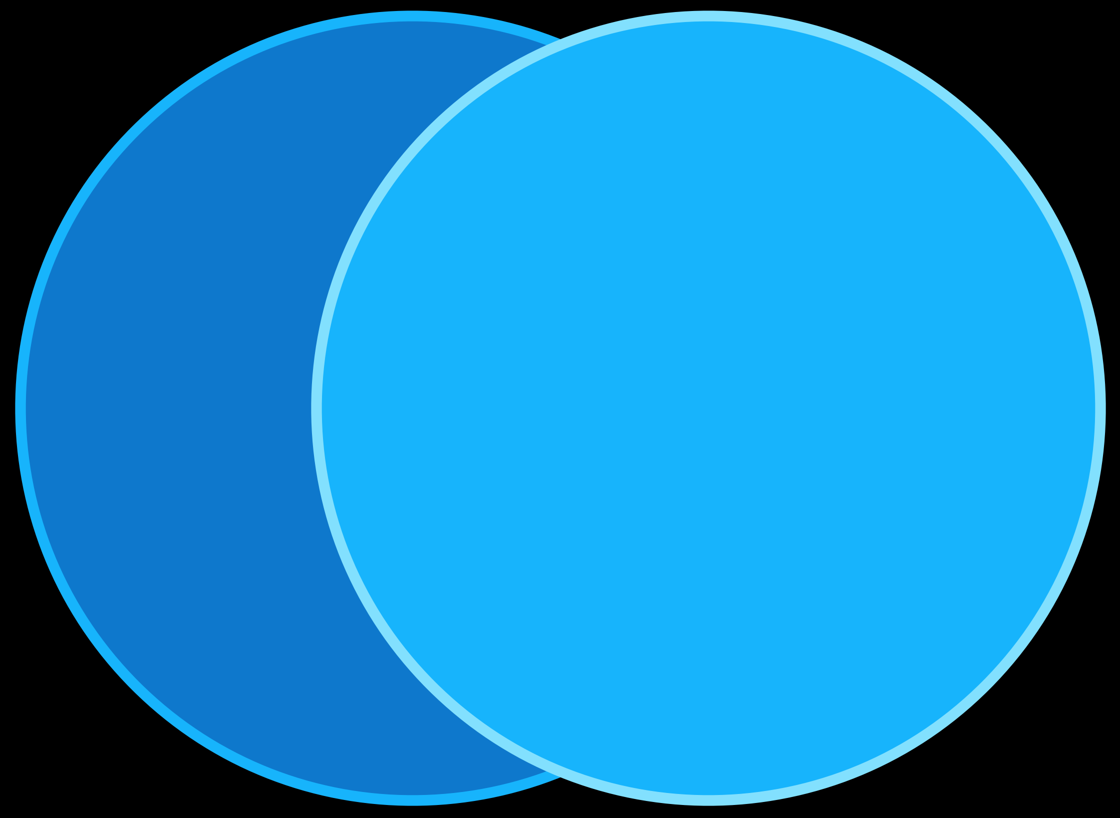
-intersectsSet: == YES

Foundation

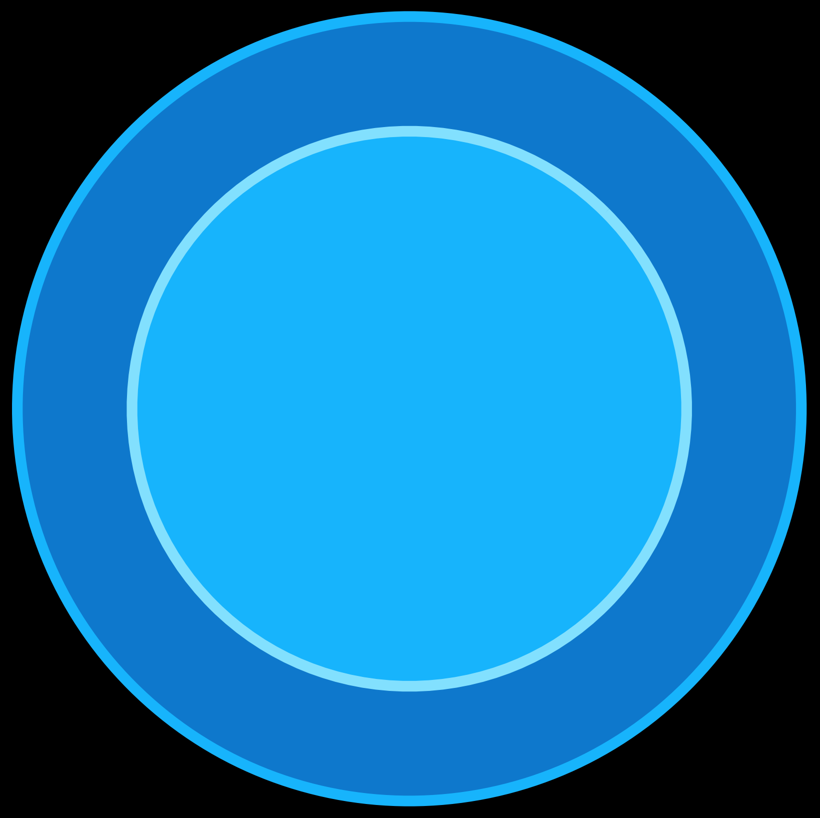
NSSet and NSOrderedSet

Tip
10

- Guaranteed uniqueness, fast membership lookup
- Membership cache
- Set calculations



-intersectsSet: == YES



-isSubsetOfSet: == YES

Foundation

NSSet and NSOrderedSet

Tip
10

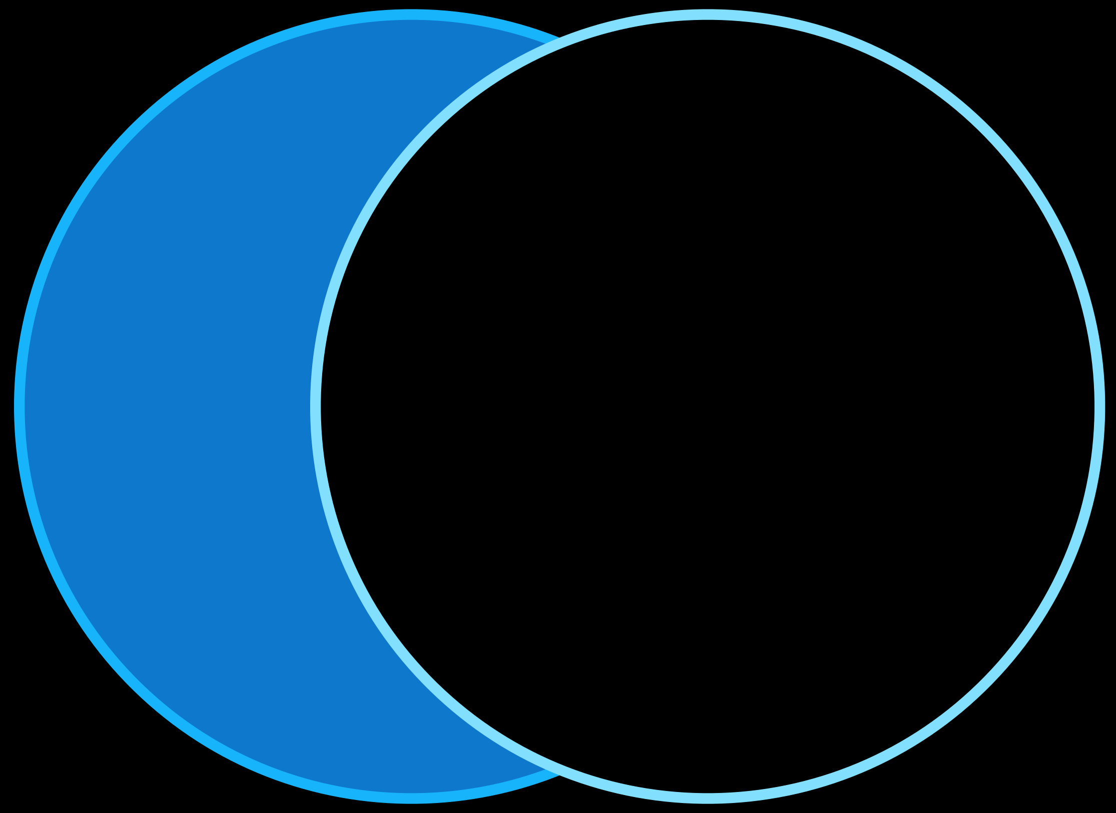
- Guaranteed uniqueness
- Membership cache
- Set calculations

Foundation

NSSet and NSOrderedSet

Tip 10

- Guaranteed uniqueness
- Membership cache
- Set calculations



-minusSet:

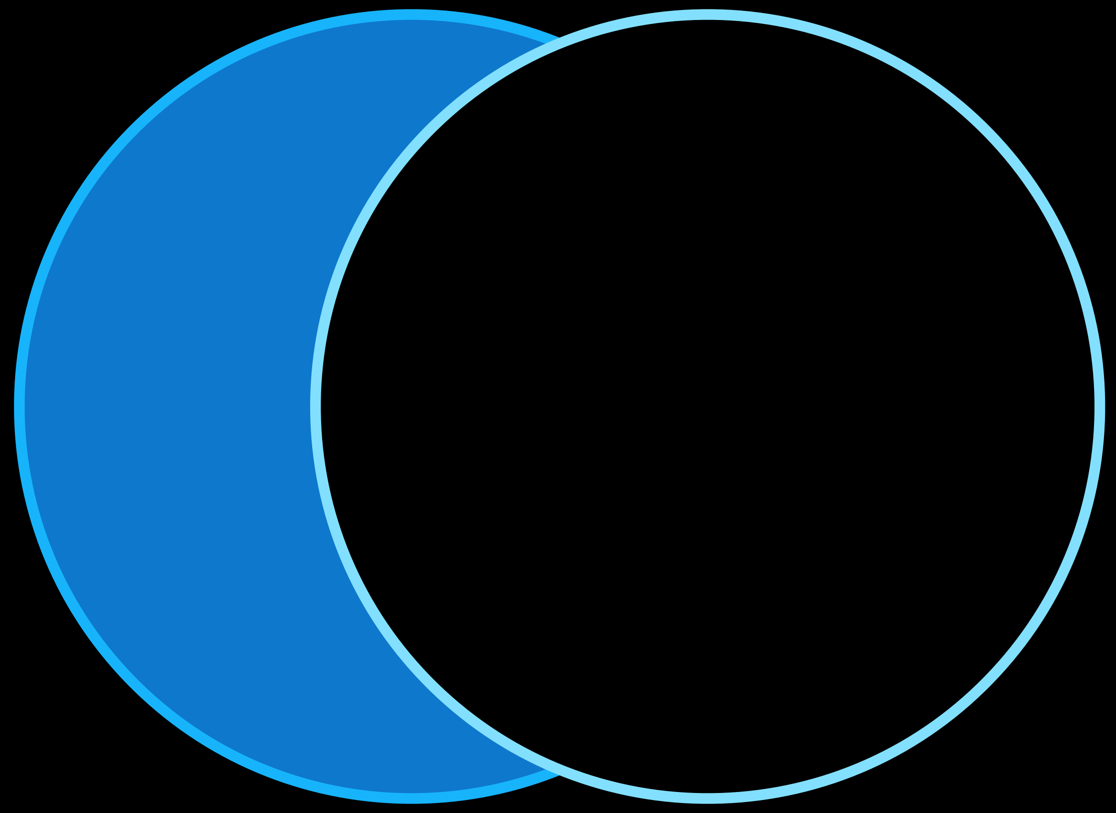
Foundation

NSSet and NSOrderedSet

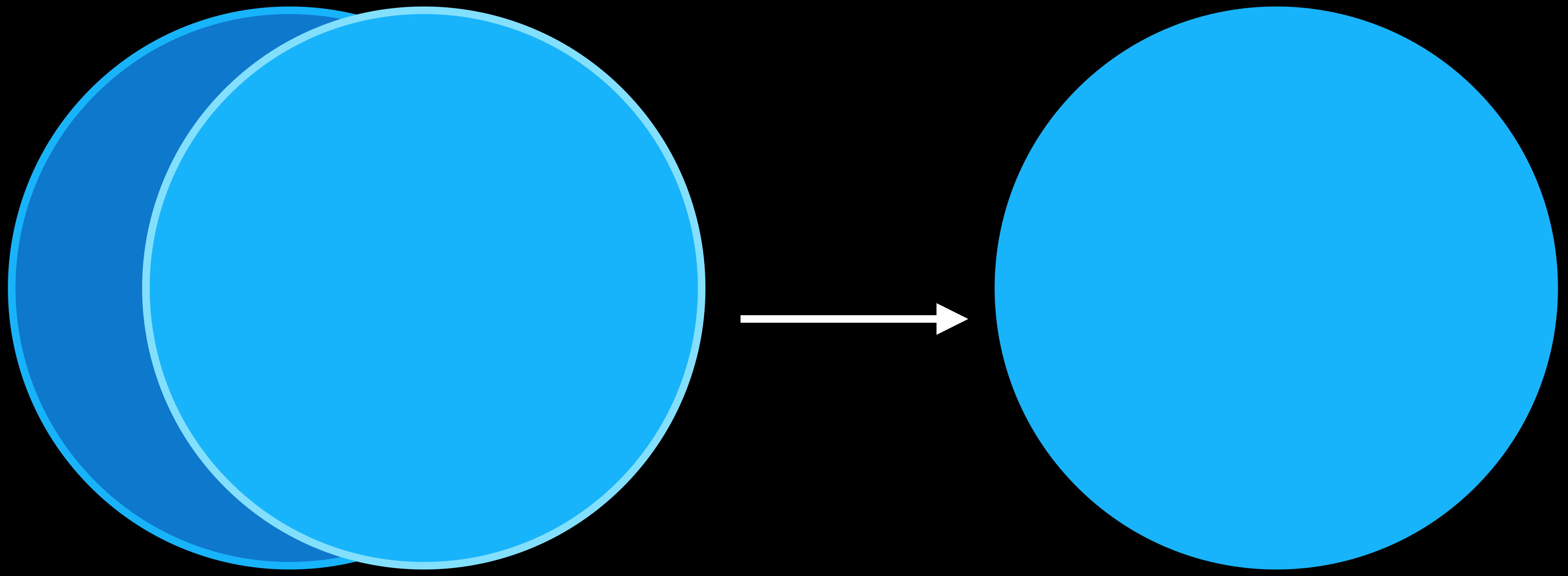
Tip

10

- Guaranteed uniqueness
- Membership cache
- Set calculations



-minusSet:



-unionSet:

Tip 10

Foundation

NSSet and NSOrderedSet

- Guaranteed uniqueness
- Membership cache
- Set calculations

Foundation

NSSet and NSOrderedSet

Tip

10

- Guaranteed uniqueness
- Membership cache
- Set calculations

Foundation

Collection tricks

Tip

11

Foundation

Collection tricks

Tip

11

- Reverse arrays quickly inline

```
NSArray *numbers = @[ @1, @2, @3 ];
NSArray *reversed = numbers.reverseObjectEnumerator.allObjects;
```

Foundation

Collection tricks

Tip 11

- Reverse arrays quickly inline

```
NSArray *numbers = @[ @1, @2, @3 ];
NSArray *reversed = numbers.reverseObjectEnumerator.allObjects;
```

- Guarantee a mutable object

```
NSArray *unknown = self.values; // may be nil
NSMutableArray *newArray = [NSMutableArray arrayWithArray:unknown];
```

Foundation

Collection tricks

Tip
11

- Reverse arrays quickly inline

```
NSArray *numbers = @[ @1, @2, @3 ];
NSArray *reversed = numbers.reverseObjectEnumerator.allObjects;
```

- Guarantee a mutable object

```
NSArray *unknown = self.values; // may be nil
NSMutableArray *newArray = [NSMutableArray arrayWithArray:unknown];
```

- Declare and enumerate different collection types

```
id<NSFastEnumeration> collection = values;
for (id object in collection) {
    ...
}
```

Foundation

NSFastEnumeration to your class

```
@implementation Manager

- (NSUInteger)countByEnumeratingWithState:(NSFastEnumerationState *)state
objects:(__unsafe_unretained id [])buffer count:(NSUInteger)len {

    return [self.subordinates
        countByEnumeratingWithState:state objects:buffer count:len];
}

@end
```

Foundation and Core Foundation

Data model tips

Mattt Thompson
NSHipster.com

NSValue

- Scalars
- Structs
  - Ranges
  - Points, sizes, and rects
- Unretained references

NSValue

Tip
12

NSValue

Tip

12

```
NSMutableArray *array = [@[] mutableCopy];
array[0] = [NSValue valueWithPoint:CGPointZero];
array[1] = [NSValue valueWithRange:NSMakeRange(3, 17)];
```

NSValue

Tip

12

```
NSMutableArray *array = [@[] mutableCopy];
array[0] = [NSValue valueWithPoint:CGPointZero];
array[1] = [NSValue valueWithRange:NSMakeRange(3, 17)];


typedef struct RGB {
     float red, green, blue;
} _RGB;
```

NSValue

Tip
12

```
NSMutableArray *array = [@[] mutableCopy];
array[0] = [NSValue valueWithPoint:CGPointZero];
array[1] = [NSValue valueWithRange:NSMakeRange(3, 17)];


typedef struct RGB {
     float red, green, blue;
} _RGB;

RGB color = {1.0f, 0.0f, 0.0f};
```

NSValue

Tip
12

```
NSMutableArray *array = [@[] mutableCopy];
array[0] = [NSValue valueWithPoint:CGPointZero];
array[1] = [NSValue valueWithRange:NSMakeRange(3, 17)];


typedef struct RGB {
     float red, green, blue;
} _RGB;

RGB color = {1.0f, 0.0f, 0.0f};
array[2] = [NSValue valueWithBytes:&color objCType:@encode(RGB)];
```

NSValue

Tip
12

- object conforms to <NSCopying>

```
NSMutableDictionary *dictionary = [@{} mutableCopy];
dictionary[object] = @42;
```

- object does not conform to <NSCopying>

```
NSMutableDictionary *dictionary = [@{} mutableCopy];
dictionary[[NSValue valueWithNonretainedObject:object]] = @42;
```

Key-Value Coding

Tip

13

```
[employee valueForKey:@"name"];
employee.name;

[employee valueForKeyPath:@"manager.name"];
employee.manager.name;
```

Key-Value Coding

Tip

13

Key-Value Coding

Tip
13

```
NSArray *words = @[@"Alpha", @"Bravo", @"Charlie"];
```

Key-Value Coding

Tip 13

```
NSArray *words = @[@"Alpha", @"Bravo", @"Charlie"];

[words valueForKey:@"uppercaseString"];
// @[@"ALPHA", @"BRAVO", @"CHARLIE"]
```

Key-Value Coding

Tip

13

```
NSArray *words = @[@"Alpha", @"Bravo", @"Charlie"];

[words valueForKey:@"uppercaseString"];
// @[@"ALPHA", @"BRAVO", @"CHARLIE"]

[words valueForKey:@"length"];
// @[@5, @5, @7]
```

Key-Value Coding

Tip

13

```
- (NSDictionary *)dictionaryWithValuesForKeys:(NSArray *)keys;
    NSMutableDictionary *dictionary = [NSMutableDictionary dictionary];
    for (NSString *key in keys) {
        dictionary[key] = [myObject valueForKey:key];
    }
}

- (void)setValuesForKeysWithDictionary:(NSDictionary *)keyedValues;
    for (NSString *key in dictionary.allKeys) {
    id value = dictionary[key];
    [myObject setValue:value forKey:key];
}
```

KVC Collection Operators

Tip

14

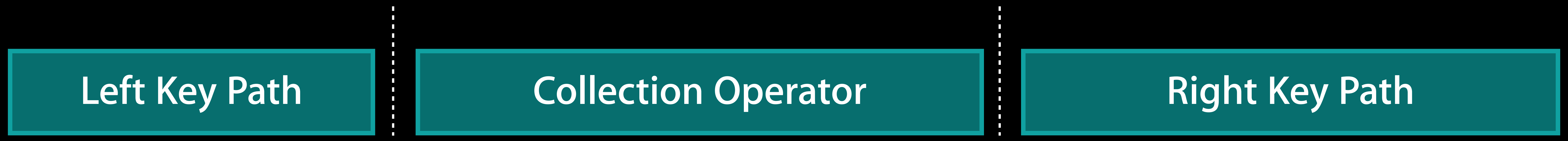
```
[employee valueForKeyPath:@"colleagues.@count"];

[staff valueForKeyPath:@"@avg.salary"];

[week valueForKeyPath:@"@max.temperature"];
```

KVC Collection Operators

- Simple collection operators
- Object operators
- Array and set operators

KVC Collection Operators

```
@"collection.@collectionOperator.keypathToProperty"
```



Simple Collection Operators

Simple Collection Operators

@count	NSNumber

Simple Collection Operators

@count	NSNumber
@sum	NSNumber
@avg	NSNumber

Simple Collection Operators

@count	NSNumber
@sum	NSNumber
@avg	NSNumber
@max	id
@min	id

Object Operators

@unionOfObjects	NSArray
@distinctUnionOfObjects	NSArray

KVC Collection Operators

Tip

14

- Remove duplicate values in array without NSSet

```
[array valueForKeyPath:@"@distinctUnionOfObjects.self"]
```

Array and Set Operators

Tip 14

@unionOfArrays	NSArray or NSSet
@distinctUnionOfArrays	NSArray or NSSet
@distinctUnionOfSets	NSArray or NSSet

Array and Set Operators

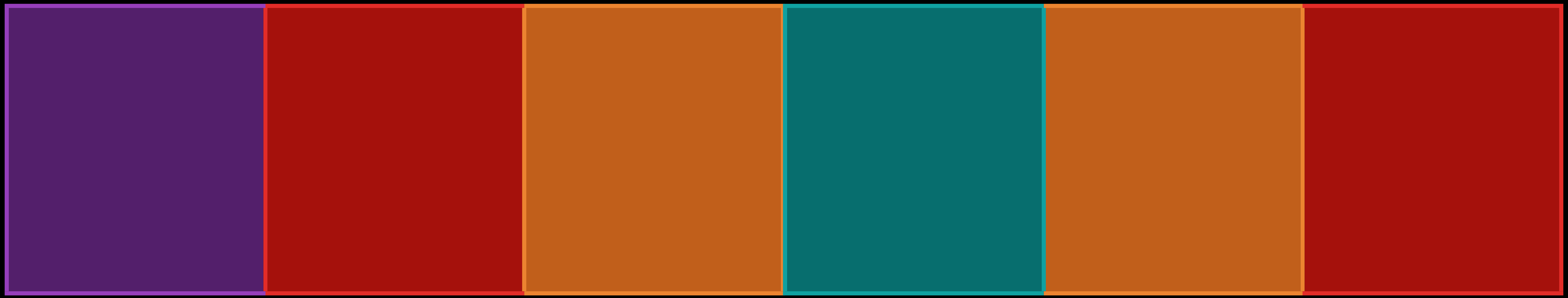
Tip
14

Array and Set Operators

Tip

14

`@unionOfArrays`



Array and Set Operators

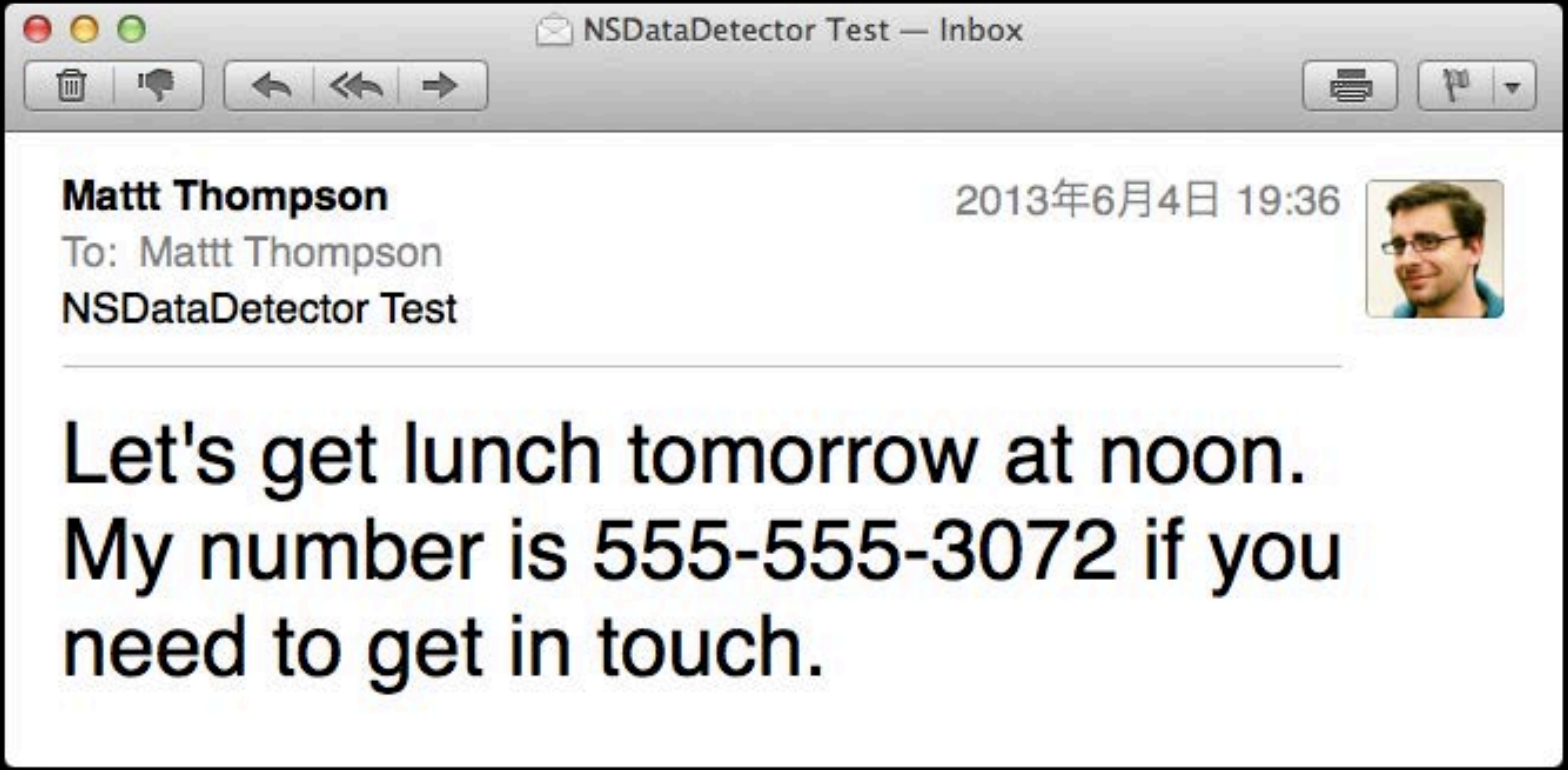
Tip

14

`@distinctUnionOfArrays`

NSDataDetector

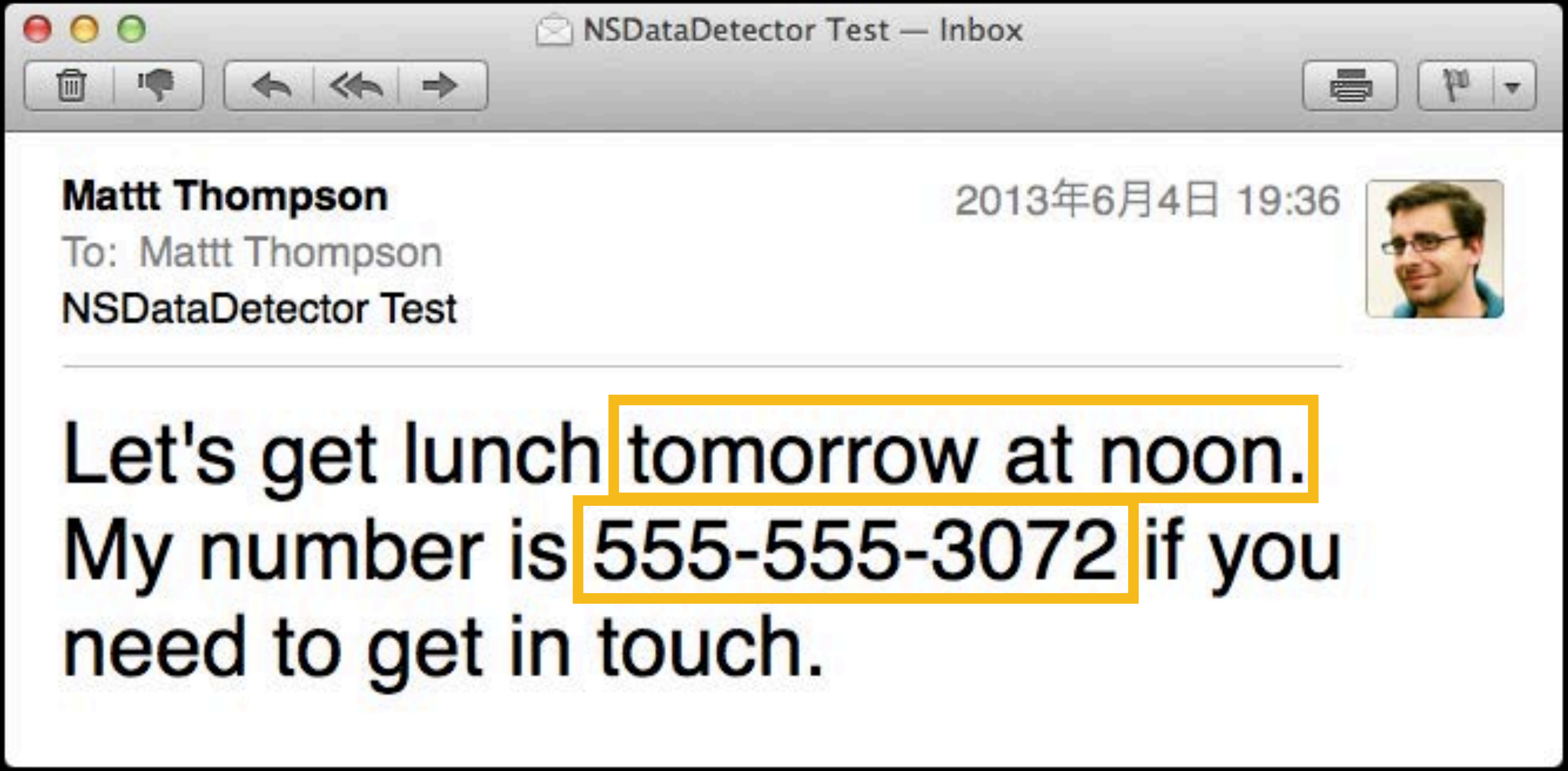
NSDataDetector


NSDataDetector Test — Inbox

Matt Thompson
2013年6月4日 19:36
To: Matt Thompson
NSDataDetector Test

Let's get lunch tomorrow at noon. My number is 555-555-3072 if you need to get in touch.

NSDataDetector


NSDataDetector Test — Inbox

Matt Thompson
To: Mattt Thompson
NSDataDetector Test

2013年6月4日 19:36

Let's get lunch tomorrow at noon. My number is 555-555-3072 if you need to get in touch.

NSDataDetector

- Dates
- Addresses
- Links
- Phone numbers
- Transit information

NSDataDetector

Tip

15

NSDataDetector

Tip

15

```
NSString *string = @"123 Main St. / (555) 555-5555";
```

NSDataDetector

Tip
15

```
NSString *string = @"123 Main St. / (555) 555-5555";

NSError *error;
```

NSDataDetector

Tip

15

```
NSString *string = @"123 Main St. / (555) 555-5555";

NSError *error;
NSDataDetector *detector = [NSDataDetector
dataDetectorWithTypes:NSTextCheckingTypeLink |
                      NSTextCheckingTypePhoneNumber
                error:&error];
```

NSDataDetector

Tip
15

```
NSString *string = @"123 Main St. / (555) 555-5555";

NSError *error;
NSDataDetector *detector = [NSDataDetector
dataDetectorWithTypes:NSTextCheckingTypeLink |
                      NSTextCheckingTypePhoneNumber
                 error:&error];

[detector enumerateMatchesInString:string
                           options:kNilOptions
                             range:NSMakeRange(0, [string length])
   usingBlock:^(NSTextCheckingResult *result, NSMatchingFlags flags,
BOOL *stop) {
      NSLog(@"Match: %@", result);
}];
```

CFStringTransform

CFStringTransform

- Strip accents and diacritics

CFStringTransform

- Strip accents and diacritics
- Name Unicode characters

CFStringTransform

- Strip accents and diacritics
- Name Unicode characters
- Encode XML hex entities

CFStringTransform

- Strip accents and diacritics
- Name Unicode characters
- Encode XML hex entities
- Transliterate between writing systems

CFStringTransform

```
Boolean CFStringTransform (
   CFMutableStringRef string,
   CFRange *range,
   CFStringRef transform,
   Boolean reverse
);
```

Strip Accents and Diacritics

Énġlišh låcks iñterêsţing diaçrïtičş

Strip Accents and Diacritics

English lacks interesting diacritics

Strip Accents and Diacritics

Tip

16

```
NSMutableString *string = [@"Énġlišh låcks iñterêsţing diaçrïtičş"
     mutableCopy];

CFStringTransform((__bridge CFMutableStringRef)(string),
    NULL, kCFStringTransformStripCombiningMarks, NO);
```

Name Unicode Characters

Name Unicode Characters

Character	Name

Name Unicode Characters

Character	Name
A	LATIN CAPITAL LETTER A

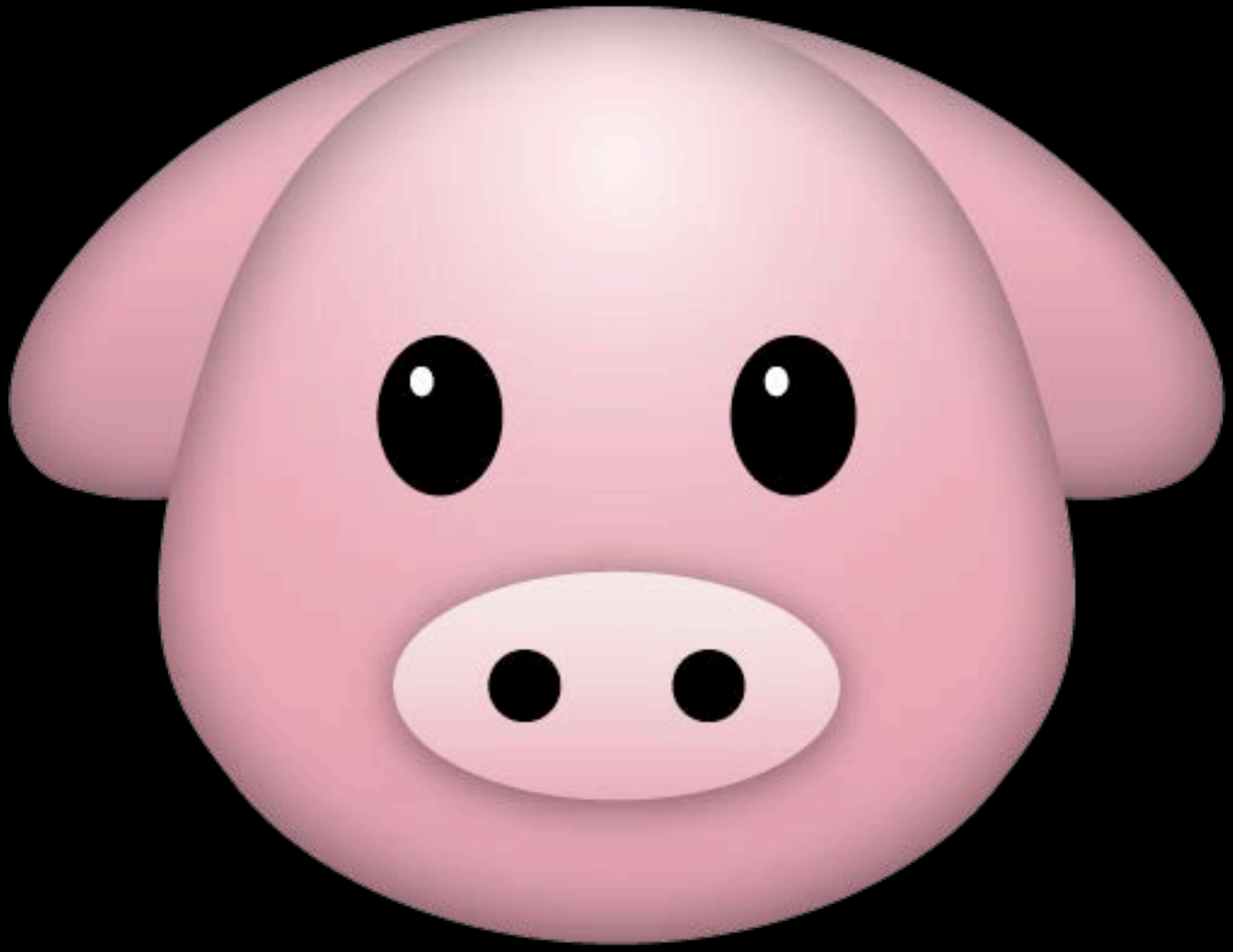
Name Unicode Characters

Character	Name
A	LATIN CAPITAL LETTER A
Å	LATIN CAPITAL LETTER A WITH RING ABOVE

Name Unicode Characters

Character	Name
A	LATIN CAPITAL LETTER A
Å	LATIN CAPITAL LETTER A WITH RING ABOVE
☃	SNOWMAN

Name Unicode Characters



Name Unicode Characters

Tip 17

```
NSMutableString *string = [@"🐷" mutableCopy];

CFStringTransform((__bridge CFMutableStringRef)(string),
    NULL, kCFStringTransformToUnicodeName, NO);
```

Name Unicode Characters

Tip
17

```
@"PIG FACE"
```

Transliterate Between Orthographies

안녕하세요

Transliterate Between Orthographies

Tip 18

```
NSMutableString *string = [@"오빤 강남스타일" mutableCopy];
CFStringTransform((__bridge CFMutableStringRef)(string),
    NULL, kCFStringTransformToLatin, NO);
```

Transliterate Between Orthographies

annyeonghaseyo

Transliteration | Transformations

Transliteration | Transformations

안녕

γειά σου

中文

שלום

مرحبا

привет

สวัสดี

Transliteration | Transformations

Transliteration | Transformations

geiá sou

şlwm

zhōng wén

annyeong

mrḥbạ

privet

s wạs dī

Transliteration | Transformations

ひらがな　　カタカナ

Transliteration | Transformations

ヒラガナ　　かたかな

Transliteration | Transformations

hiragana katakana

“On OS X v10.4 and later, with CFStringTransform you can also use any valid ICU transform ID defined in the ICU User Guide for Transforms.”

Apple Developer Documentation

Transliteration | Transformations

Transliteration | Transformations

gɹiːtɪŋs

હેલ્લો

హంలో

მიესალმები

नमस्ते

হ্যালো

Բարև Ձեզ

Normalizing Input

hello! こんにちは! สวัสดี! مرحبا! 您好!

Normalizing Input

Tip

19

Normalizing Input

Tip
19

```
CFMutableStringRef string;
string = (__bridge CFMutableStringRef)[@"Hello! こんにちは! สวัสดี! مرحبا! 您好!"
    mutableCopy];
```

Normalizing Input

Tip
19

```
CFMutableStringRef string;
string = (__bridge CFMutableStringRef)[@"Hello! こんにちは! สวัสดี! مرحبا! 您好!"
    mutableCopy];

// Hello! kon'nichiha! swạsdī! mrḥbạ! nín hǎo!
```

Normalizing Input

Tip 19

```
CFMutableStringRef string;
string = (__bridge CFMutableStringRef)[@"Hello! こんにちは! สวัสดี! مرحبا! 您好!"
    mutableCopy];

// Hello! kon'nichiha! swạsdī! mrḥbạ! nín hǎo!
CFStringTransform(string, NULL, kCFStringTransformToLatin, NO);
```

Normalizing Input

```
CFMutableStringRef string;
string = (__bridge CFMutableStringRef)[@"Hello! こんにちは! สวัสดี! مرحبا! 您好!"
    mutableCopy];

// Hello! kon'nichiha! swạsdī! mrḥbạ! nín hǎo!
CFStringTransform(string, NULL, kCFStringTransformToLatin, NO);

// Hello! kon'nichiha! swasdi! mrhba! nin hao!
```

Normalizing Input

Tip
19

```
CFMutableStringRef string;
string = (__bridge CFMutableStringRef)[@"Hello! こんにちは! สวัสดี! مرحبا! 您好!"
    mutableCopy];

// Hello! kon'nichiha! swạsdī! mrḥbạ! nín hǎo!
CFStringTransform(string, NULL, kCFStringTransformToLatin, NO);

// Hello! kon'nichiha! swasdi! mrhba! nin hao!
CFStringTransform(string, NULL, kCFStringTransformStripCombiningMarks, NO);
```

Normalizing Input

Tip
19

```
CFMutableStringRef string;
string = (__bridge CFMutableStringRef)[@"Hello! こんにちは! สวัสดี! مرحبا! 您好!"
    mutableCopy];

// Hello! kon'nichiha! swạsdī! mrḥbạ! nín hǎo!
CFStringTransform(string, NULL, kCFStringTransformToLatin, NO);

// Hello! kon'nichiha! swasdi! mrhba! nin hao!
CFStringTransform(string, NULL, kCFStringTransformStripCombiningMarks, NO);

// hello! kon'nichiha! swasdi! mrhba! nin hao!
```

Normalizing Input

Tip 19

```
CFMutableStringRef string;
string = (__bridge CFMutableStringRef)[@"Hello! こんにちは! สวัสดี! مرحبا! 您好!"
    mutableCopy];

// Hello! kon'nichiha! swạsdī! mrḥbạ! nín hǎo!
CFStringTransform(string, NULL, kCFStringTransformToLatin, NO);

// Hello! kon'nichiha! swasdi! mrhba! nin hao!
CFStringTransform(string, NULL, kCFStringTransformStripCombiningMarks, NO);

// hello! kon'nichiha! swasdi! mrhba! nin hao!
CFStringLowercase(string, NULL);
```

Normalizing Input

Tip

19

Normalizing Input

Tip
19

```
NSMutableArray *mutableWords = [NSMutableArray array];
```

Normalizing Input

Tip
19

```
NSMutableArray *mutableWords = [NSMutableArray array];
[string enumerateLinguisticTagsInRange:NSMakeRange(0, string.length)
```

Normalizing Input

Tip
19

```
NSMutableArray *mutableWords = [NSMutableArray array];
[string enumerateLinguisticTagsInRange:NSMakeRange(0, string.length)
                                scheme:NSLinguisticTagSchemeTokenType
```

Normalizing Input

```
NSMutableArray *mutableWords = [NSMutableArray array];
[string enumerateLinguisticTagsInRange:NSMakeRange(0, string.length)
                                 scheme:NSLinguisticTagSchemeTokenType
                                options:kNilOptions
```

Normalizing Input

Tip

19

```
NSMutableArray *mutableWords = [NSMutableArray array];
[string enumerateLinguisticTagsInRange:NSMakeRange(0, string.length)
                                 scheme:NSLinguisticTagSchemeTokenType
                                options:kNilOptions
                           orthography:nil
```

Normalizing Input

Tip

19

```
NSMutableArray *mutableWords = [NSMutableArray array];
[string enumerateLinguisticTagsInRange:NSMakeRange(0, string.length)
                                scheme:NSLinguisticTagSchemeTokenType
                               options:kNilOptions
                           orthography:nil
                            usingBlock:
  ^(NSString *tag, NSRange tokenRange, NSRange sentenceRange, BOOL *stop)
  {
       if ([tag isEqualToString:NSLinguisticTagWord]) {
        [mutableWords addObject:[string substringWithRange:tokenRange]];
    }
}];
```

Normalizing Input

hello! こんにちは*!* สวัสดี*!* مرحبا*!* 您好*!*

Normalizing Input

hello! こんにちは*!* สวัสดี*!* مرحبا*!* 您好*!*

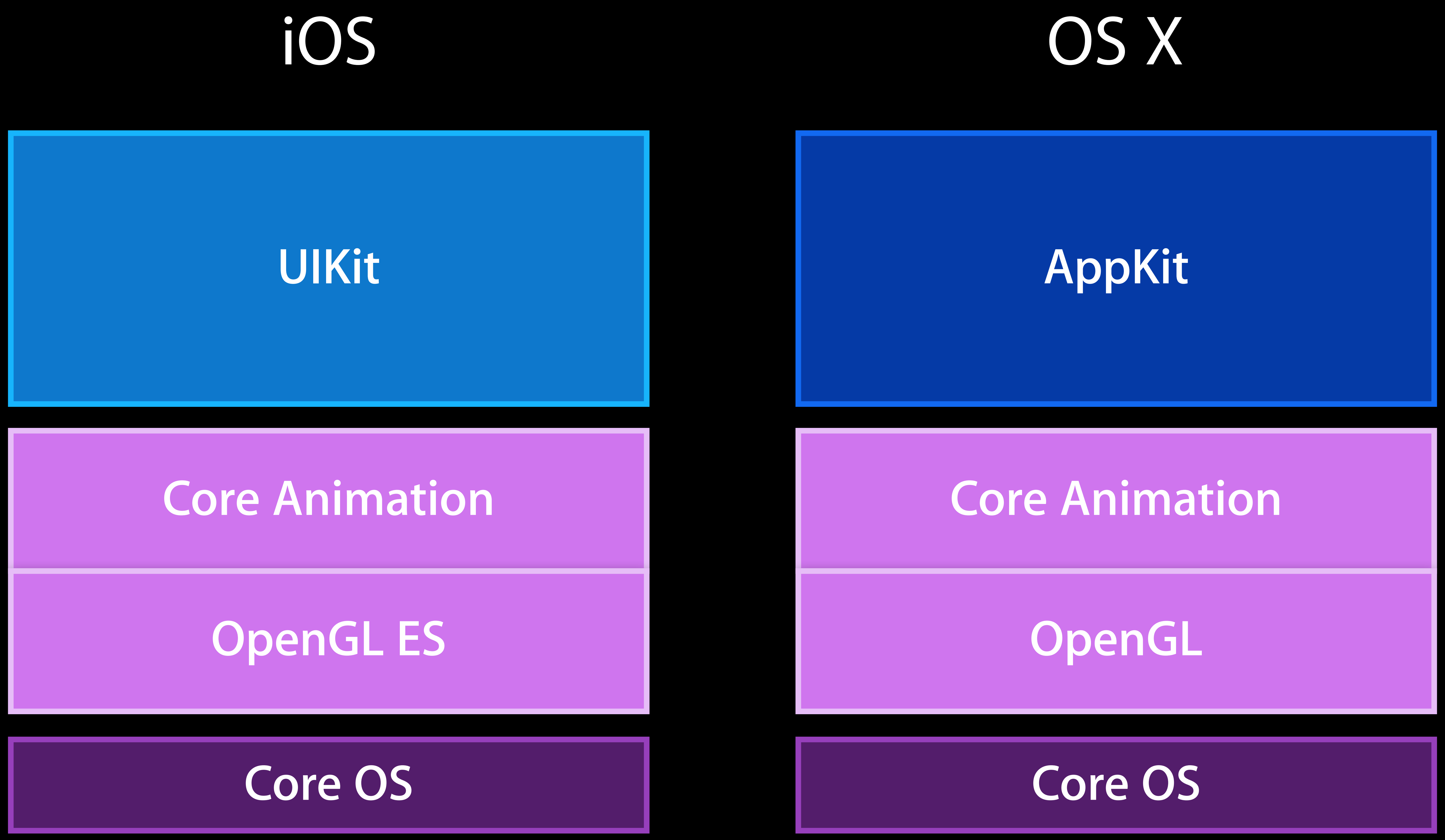
hello, kon'nichiha, swasdi, mrhba, nin, hao

Core Animation

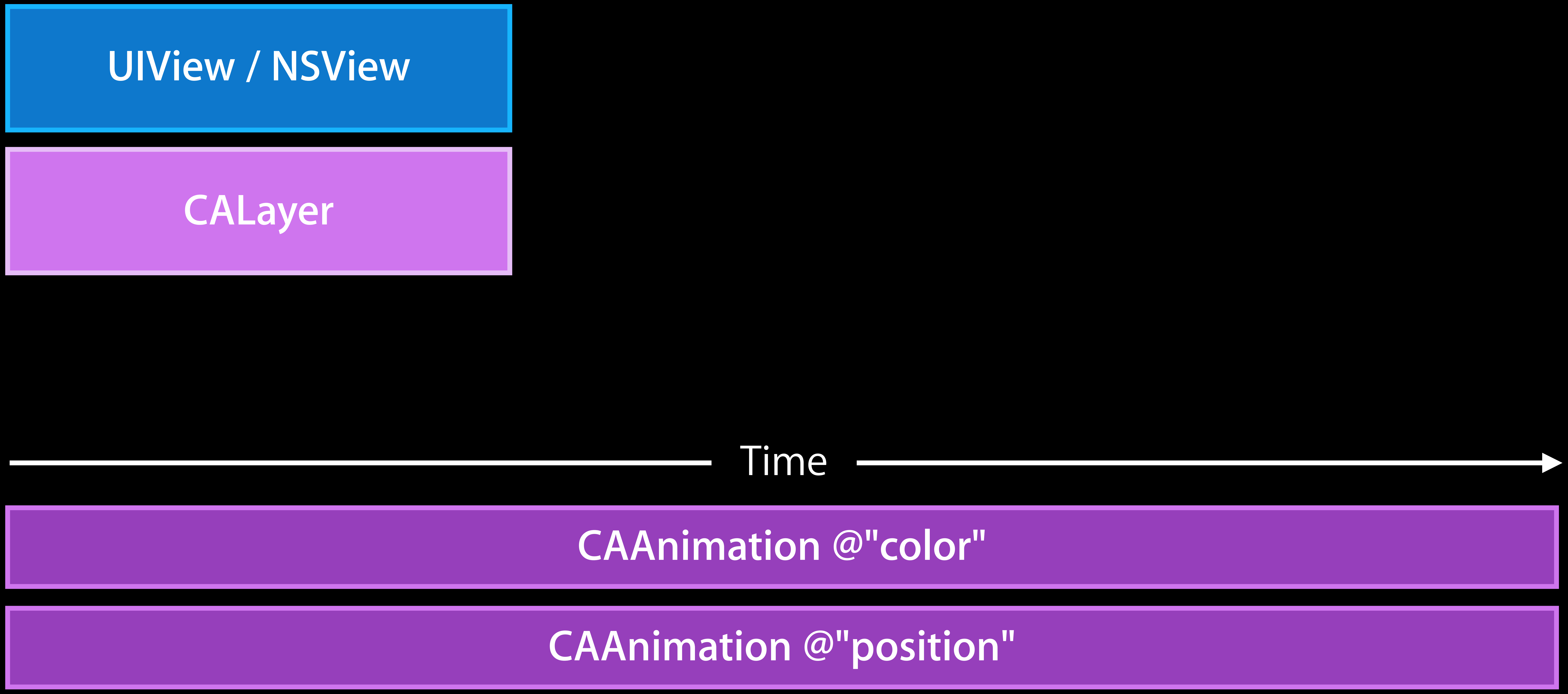
Animation tips

Scott Stevenson
Software Engineer

Core Animation

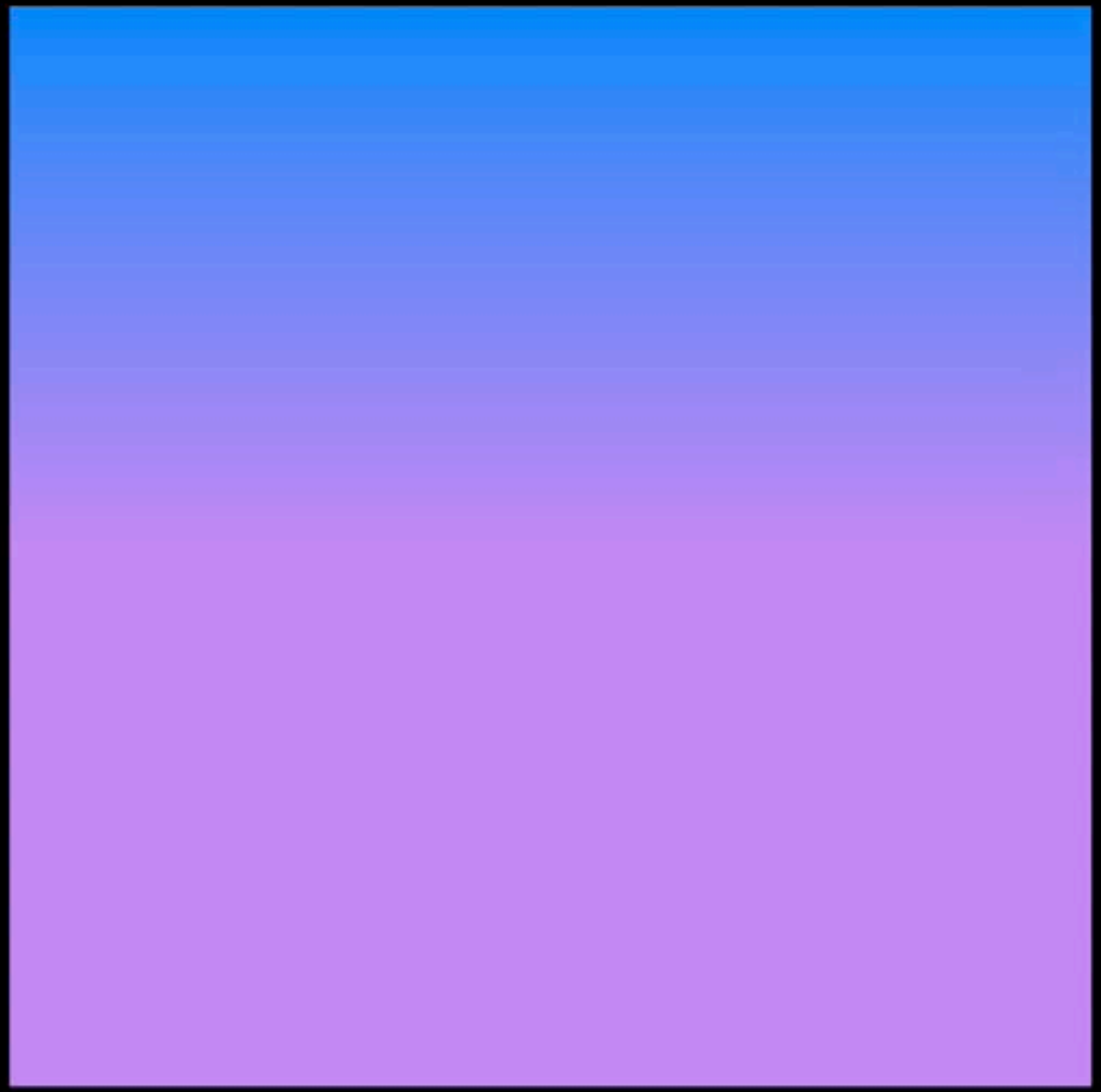


Core Animation



Core Animation
CAGradientLayer

Tip
20



Core Animation

CAGradientLayer

Tip
20

```
CAGradientLayer *gradient = [CAGradientLayer layer];
gradient.frame = CGRectMake(150, 250, 500, 500);

UIColor *c1 = [UIColor colorWithRed:0.09 green:0.70 blue:0.98 alpha:1.0];
UIColor *c2 = [UIColor colorWithRed:0.07 green:0.41 blue:0.95 alpha:1.0];
UIColor *c3 = [UIColor colorWithRed:0.81 green:0.46 blue:0.93 alpha:1.0];
gradient.colors = @[(id)c2.CGColor, (id)c3.CGColor, (id)c3.CGColor];

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"colors"];
anim.toValue = @[(id)c1.CGColor, (id)c2.CGColor, (id)c2.CGColor];
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"colors"];

[self.view.layer addSublayer:gradient];
```

Core Animation

CAGradientLayer

Tip

20

```
CAGradientLayer *gradient = [CAGradientLayer layer];
gradient.frame = CGRectMake(150, 250, 500, 500);

UIColor *c1 = [UIColor colorWithRed:0.09 green:0.70 blue:0.98 alpha:1.0];
UIColor *c2 = [UIColor colorWithRed:0.07 green:0.41 blue:0.95 alpha:1.0];
UIColor *c3 = [UIColor colorWithRed:0.81 green:0.46 blue:0.93 alpha:1.0];
gradient.colors = @[(id)c2.CGColor, (id)c3.CGColor, (id)c3.CGColor];

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"colors"];
anim.toValue = @[(id)c1.CGColor, (id)c2.CGColor, (id)c2.CGColor];
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"colors"];

[self.view.layer addSublayer:gradient];
```

Core Animation

CAGradientLayer

Tip
20

```
CAGradientLayer *gradient = [CAGradientLayer layer];
gradient.frame = CGRectMake(150, 250, 500, 500);

UIColor *c1 = [UIColor colorWithRed:0.09 green:0.70 blue:0.98 alpha:1.0];
UIColor *c2 = [UIColor colorWithRed:0.07 green:0.41 blue:0.95 alpha:1.0];
UIColor *c3 = [UIColor colorWithRed:0.81 green:0.46 blue:0.93 alpha:1.0];
gradient.colors = @[(id)c2.CGColor, (id)c3.CGColor, (id)c3.CGColor];

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"colors"];
anim.toValue = @[(id)c1.CGColor, (id)c2.CGColor, (id)c2.CGColor];
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"colors"];

[self.view.layer addSublayer:gradient];
```

Core Animation

CAGradientLayer

Tip 20

```
CAGradientLayer *gradient = [CAGradientLayer layer];
gradient.frame = CGRectMake(150, 250, 500, 500);

UIColor *c1 = [UIColor colorWithRed:0.09 green:0.70 blue:0.98 alpha:1.0];
UIColor *c2 = [UIColor colorWithRed:0.07 green:0.41 blue:0.95 alpha:1.0];
UIColor *c3 = [UIColor colorWithRed:0.81 green:0.46 blue:0.93 alpha:1.0];
gradient.colors = @[(id)c2.CGColor, (id)c3.CGColor, (id)c3.CGColor];

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"colors"];
anim.toValue = @[(id)c1.CGColor, (id)c2.CGColor, (id)c2.CGColor];
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"colors"];

[self.view.layer addSublayer:gradient];
```

Core Animation

CAGradientLayer

Tip

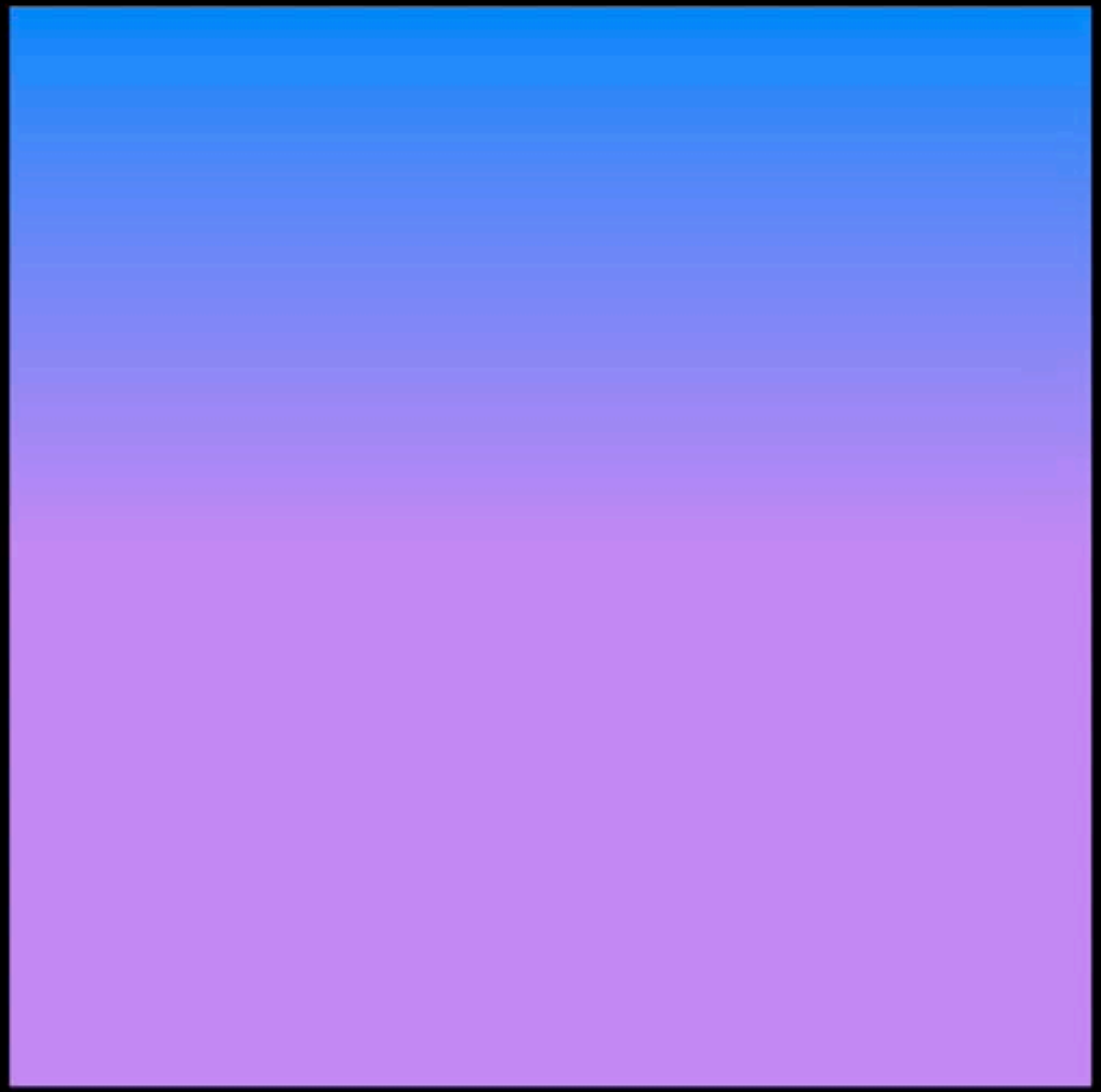
20

```
CAGradientLayer *gradient = [CAGradientLayer layer];
gradient.frame = CGRectMake(150, 250, 500, 500);

UIColor *c1 = [UIColor colorWithRed:0.09 green:0.70 blue:0.98 alpha:1.0];
UIColor *c2 = [UIColor colorWithRed:0.07 green:0.41 blue:0.95 alpha:1.0];
UIColor *c3 = [UIColor colorWithRed:0.81 green:0.46 blue:0.93 alpha:1.0];
gradient.colors = @[(id)c2.CGColor, (id)c3.CGColor, (id)c3.CGColor];

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"colors"];
anim.toValue = @[(id)c1.CGColor, (id)c2.CGColor, (id)c2.CGColor];
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"colors"];

[self.view.layer addSublayer:gradient];
```

Core Animation
CAGradientLayer

Tip

20



Core Animation
CAShapeLayer

Tip
21

Core Animation

CAShapeLayer

Tip
21

```
CAShapeLayer *shapeLayer = [CAShapeLayer layer];
shapeLayer.frame = CGRectMake(150, 250, 500, 500);

UIBezierPath* path1 = [UIBezierPath bezierPath];
// ... add points to shape
shapeLayer.path = path1.CGPath;

UIBezierPath* path2 = [UIBezierPath bezierPath];
// ... add points to shape

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"path"];
anim.toValue = (id)path2.CGPath;
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"path"];

[self.view.layer addSublayer:gradient];
```

Core Animation

CAShapeLayer

Tip 21

```
CAShapeLayer *shapeLayer = [CAShapeLayer layer];
shapeLayer.frame = CGRectMake(150, 250, 500, 500);

UIBezierPath* path1 = [UIBezierPath bezierPath];
// ... add points to shape
shapeLayer.path = path1.CGPath;

UIBezierPath* path2 = [UIBezierPath bezierPath];
// ... add points to shape

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"path"];
anim.toValue = (id)path2.CGPath;
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"path"];

[self.view.layer addSublayer:gradient];
```

Core Animation

CAShapeLayer

Tip
21

```
CAShapeLayer *shapeLayer = [CAShapeLayer layer];
shapeLayer.frame = CGRectMake(150, 250, 500, 500);

UIBezierPath* path1 = [UIBezierPath bezierPath];
// ... add points to shape
shapeLayer.path = path1.CGPath;

UIBezierPath* path2 = [UIBezierPath bezierPath];
// ... add points to shape

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"path"];
anim.toValue = (id)path2.CGPath;
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"path"];

[self.view.layer addSublayer:gradient];
```

Core Animation

CAShapeLayer

Tip

21

```
CAShapeLayer *shapeLayer = [CAShapeLayer layer];
shapeLayer.frame = CGRectMake(150, 250, 500, 500);

UIBezierPath* path1 = [UIBezierPath bezierPath];
// ... add points to shape
shapeLayer.path = path1.CGPath;

UIBezierPath* path2 = [UIBezierPath bezierPath];
// ... add points to shape

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"path"];
anim.toValue = (id)path2.CGPath;
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"path"];

[self.view.layer addSublayer:gradient];
```

Core Animation
CAShapeLayer

Tip 21

```
CAShapeLayer *shapeLayer = [CAShapeLayer layer];
shapeLayer.frame = CGRectMake(150, 250, 500, 500);

UIBezierPath* path1 = [UIBezierPath bezierPath];
// ... add points to shape
shapeLayer.path = path1.CGPath;

UIBezierPath* path2 = [UIBezierPath bezierPath];
// ... add points to shape

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"path"];
anim.toValue = (id)path2.CGPath;
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"path"];

[self.view.layer addSublayer:gradient];
```

Core Animation

CAShapeLayer

Tip

21

```
CAShapeLayer *shapeLayer = [CAShapeLayer layer];
shapeLayer.frame = CGRectMake(150, 250, 500, 500);

UIBezierPath* path1 = [UIBezierPath bezierPath];
// ... add points to shape
shapeLayer.path = path1.CGPath;

UIBezierPath* path2 = [UIBezierPath bezierPath];
// ... add points to shape

CABasicAnimation *anim = [CABasicAnimation animationWithKeyPath:@"path"];
anim.toValue = (id)path2.CGPath;
anim.duration = 4.0;
anim.autoreverses = YES;
anim.repeatCount = 1e100;
[gradient addAnimation:anim forKey:@"path"];

[self.view.layer addSublayer:gradient];
```

Core Animation
CAShapeLayer

Tip
21

Core Animation

CAShapeLayer + CAGradientLayer = ?

Tip

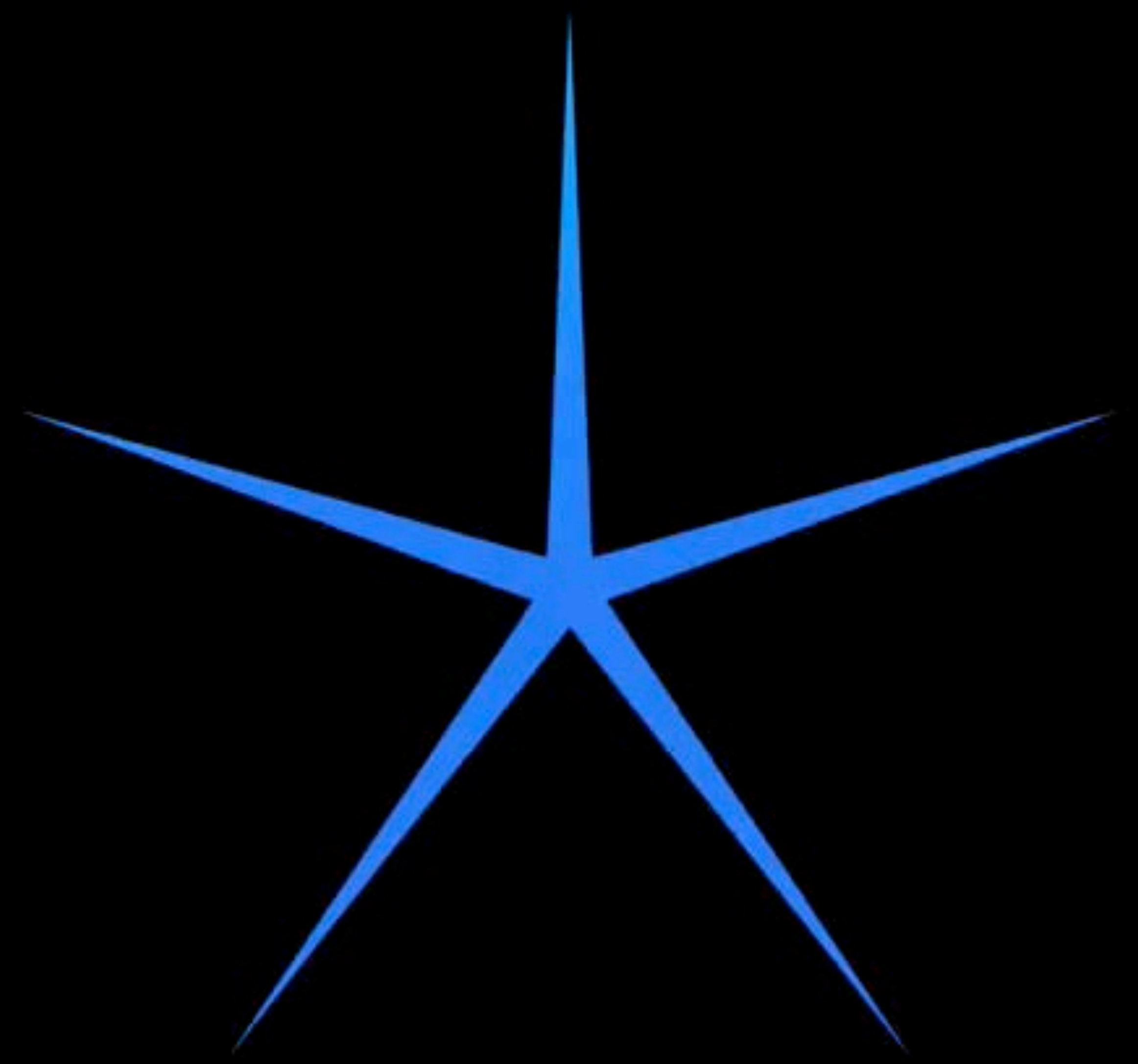
22

Core Animation

CAShapeLayer + CAGradientLayer = ?

Tip

22

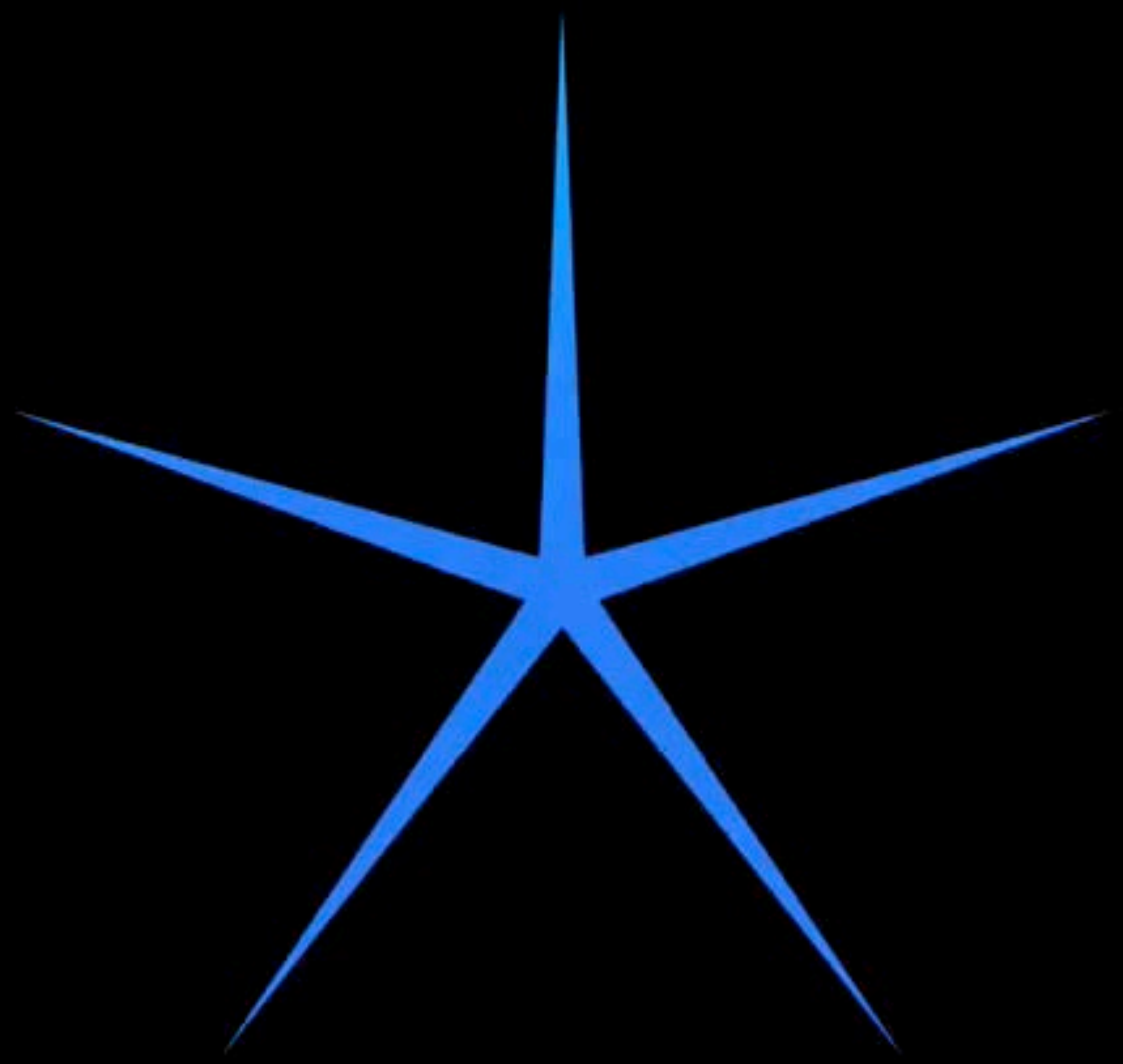


Core Animation

CAShapeLayer + CAGradientLayer = ?

Tip

22

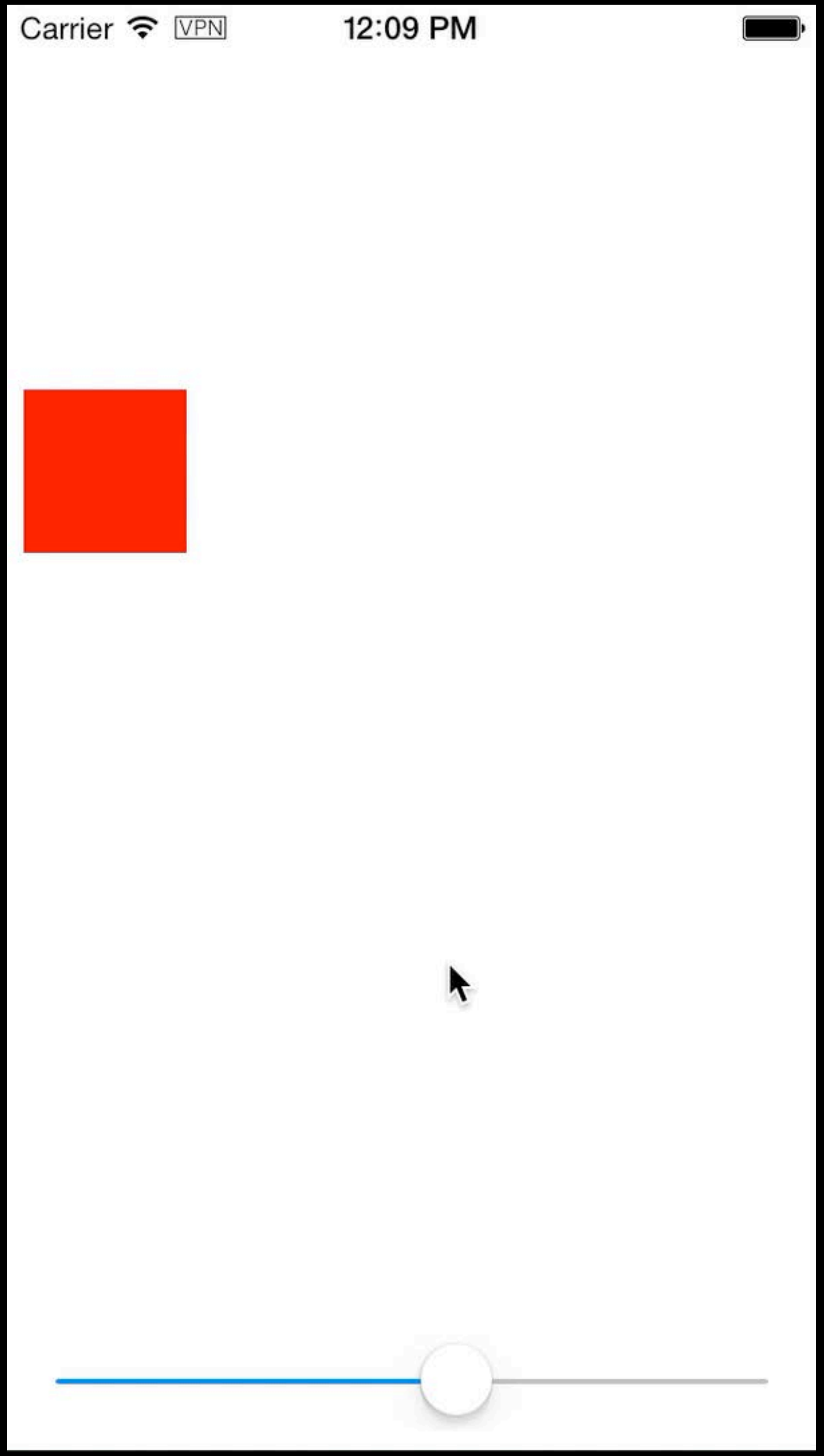


```
self.gradientLayer.mask = shapeLayer;
```

Core Animation

Interactive animation

Tip
22



Core Animation

CAMediaTiming for interactive animations

Tip
22

```
#import "MyViewController.h"
#import <QuartzCore/QuartzCore.h>

@interface MyViewController ()

@property (nonatomic, weak) IBOutlet UIView *squareContainer;
@property (nonatomic, weak) IBOutlet UIView *square;
@property (nonatomic, weak) IBOutlet UISlider *slider;

@end
```

Core Animation

CAMediaTiming for interactive animations

Tip
22

```
- (void)viewDidAppear:(BOOL)animated {
    [super viewDidAppear:animated];
    self.square.layer.speed = 0;

    CGRect rect = self.squareContainer.bounds;
    rect.size.height = CGRectGetMinY(self.slider.frame);
    CGFloat dx = CGRectGetWidth(self.square.frame)/2.0;
    CGFloat dy = CGRectGetHeight(self.square.frame)/2.0;
    CGRect r = CGRectInset(rect, dx, dy);

    CAKeyframeAnimation *animation = [CAKeyframeAnimation
        animationWithKeyPath:@"position"];
    animation.path = [[UIBezierPath bezierPathWithOvalInRect:r] CGPath];
    animation.calculationMode = kCAAnimationPaced;
    animation.speed = 0.25;
    [self.square.layer addAnimation:animation forKey:@"position"];
}
```

Core Animation

CAMediaTiming for interactive animations

Tip

22

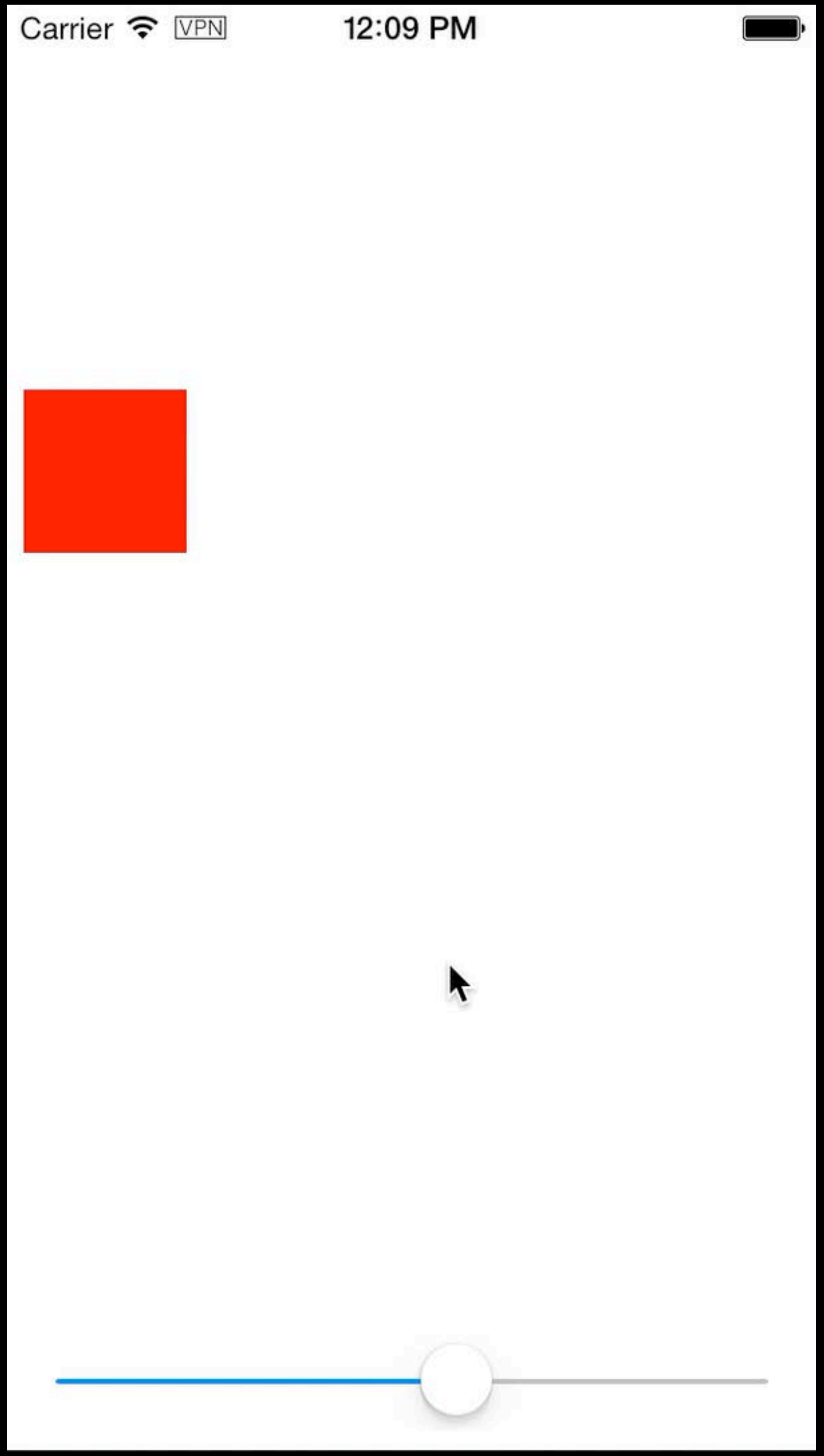
```
- (IBAction)sliderChanged:(id)sender {
    self.square.layer.timeOffset = self.slider.value;
}
```

Core Animation

Interactive animation

Tip

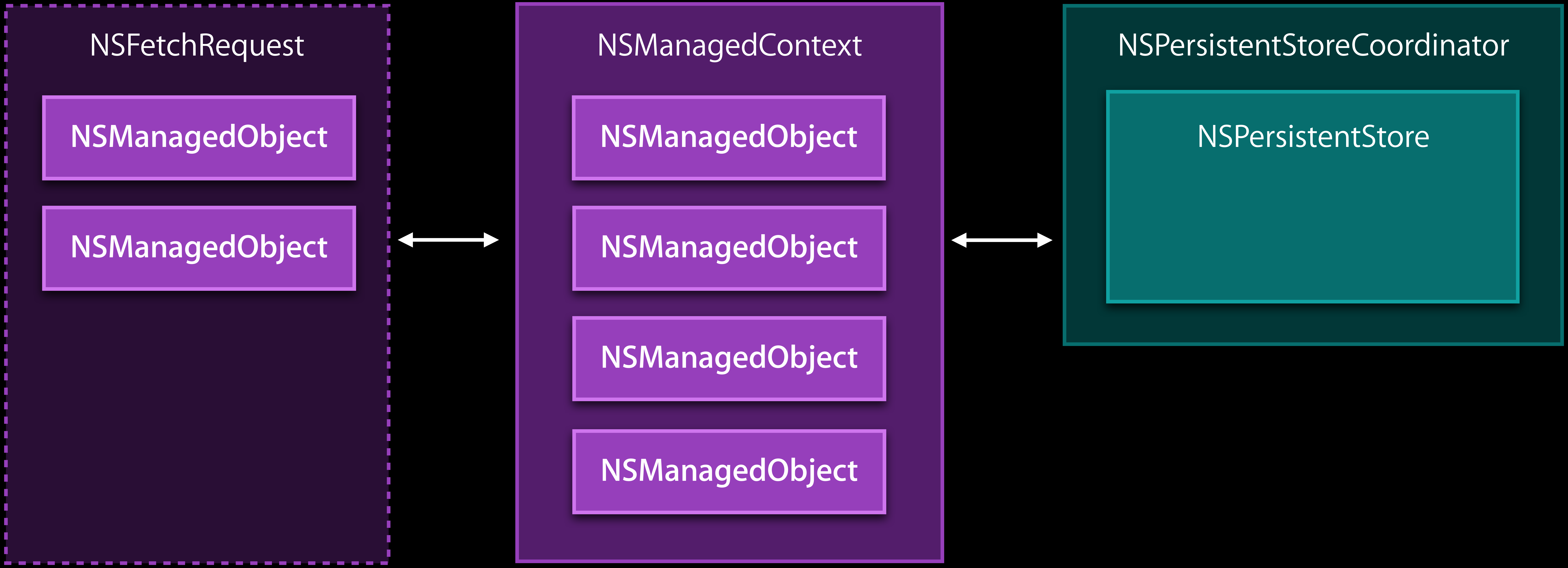
22



Core Data

Performance tips

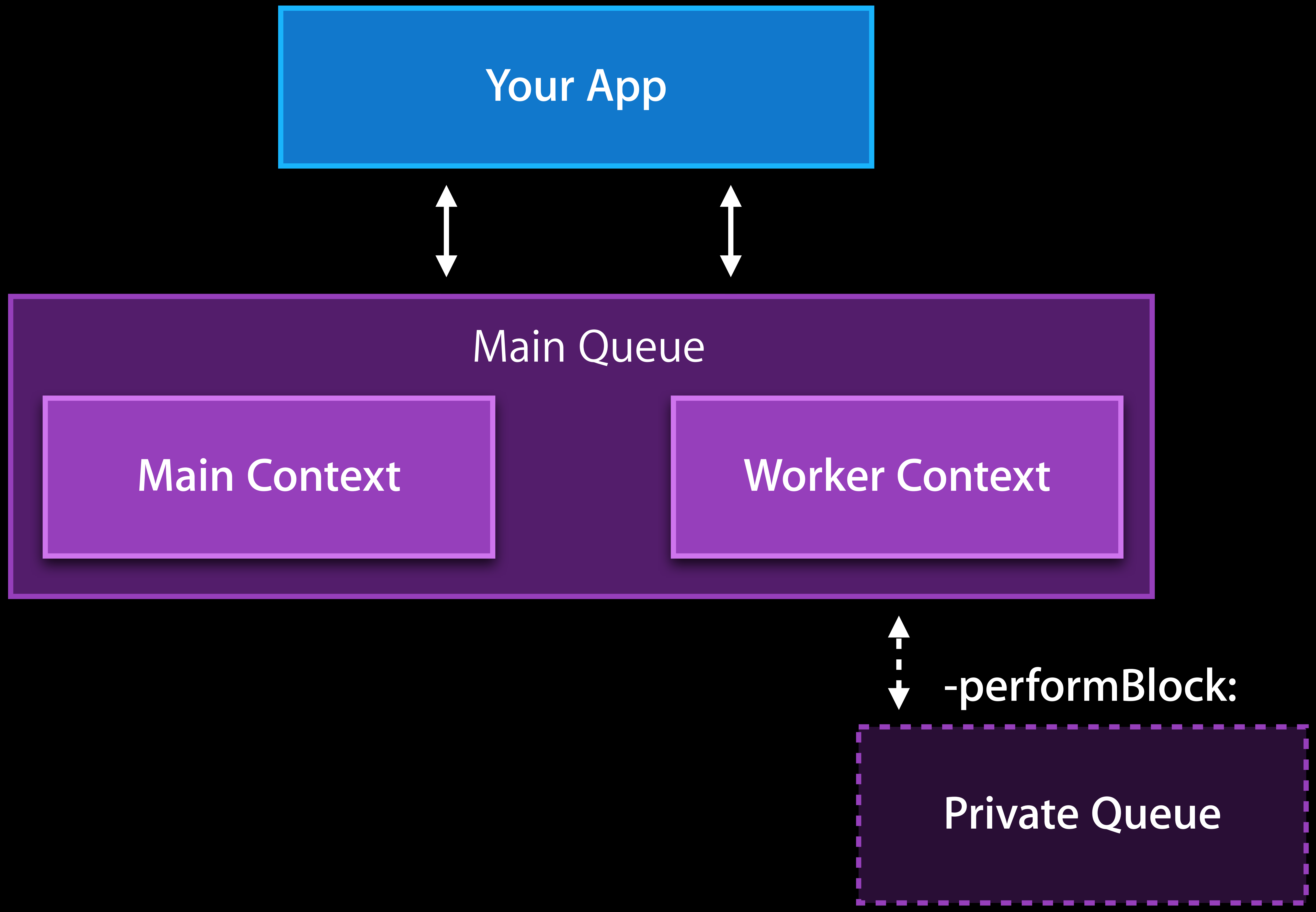
Core Data


NSFetchRequest
NSManagedObject
NSManagedObject
NSManagedContext
NSManagedObject
NSManagedObject
NSManagedObject
NSManagedObject
NSPersistentStoreCoordinator
NSPersistentStore

Core Data

NSPrivateQueueConcurrencyType

Tip 23



Core Data

Private Queue

Tip 23

```
NSManagedObjectContext *bgContext;
bgContext = [[NSManagedObjectContext alloc] initWithConcurrencyType:
                                    NSPrivateQueueConcurrencyType];

[context performBlock:^{
    // (add, remove, change objects.)
    saveCompleted = [context save:& saveError];
}];
```

Core Data

Really fast fetches

Tip 24

- Only specific properties

```
NSFetchRequest
fetch.propertiesToFetch = @[@"name", @"phone"];
```

- Only the raw values

```
fetch.resultType = NSDictionaryResultType
```

- Only the object id

```
fetch.resultType = NSManagedObjectIDResultType
```

- Only the count

```
fetch.resultType = NSCountResultType
```

Core Data

Really fast fetches

Tip

25

- Work in groups of objects

```
NSFetchRequest
fetch.fetchBatchSize = 100;
```

- Reduce cross-entity fetches

```
fetch.relationshipKeyPathsForPrefetching = @[ @"artist", @"catalog" ];
```

Core Data

Really fast fetches

Tip

25

- Work in groups of objects

```
NSFetchRequest
fetch.fetchBatchSize = 100;
```

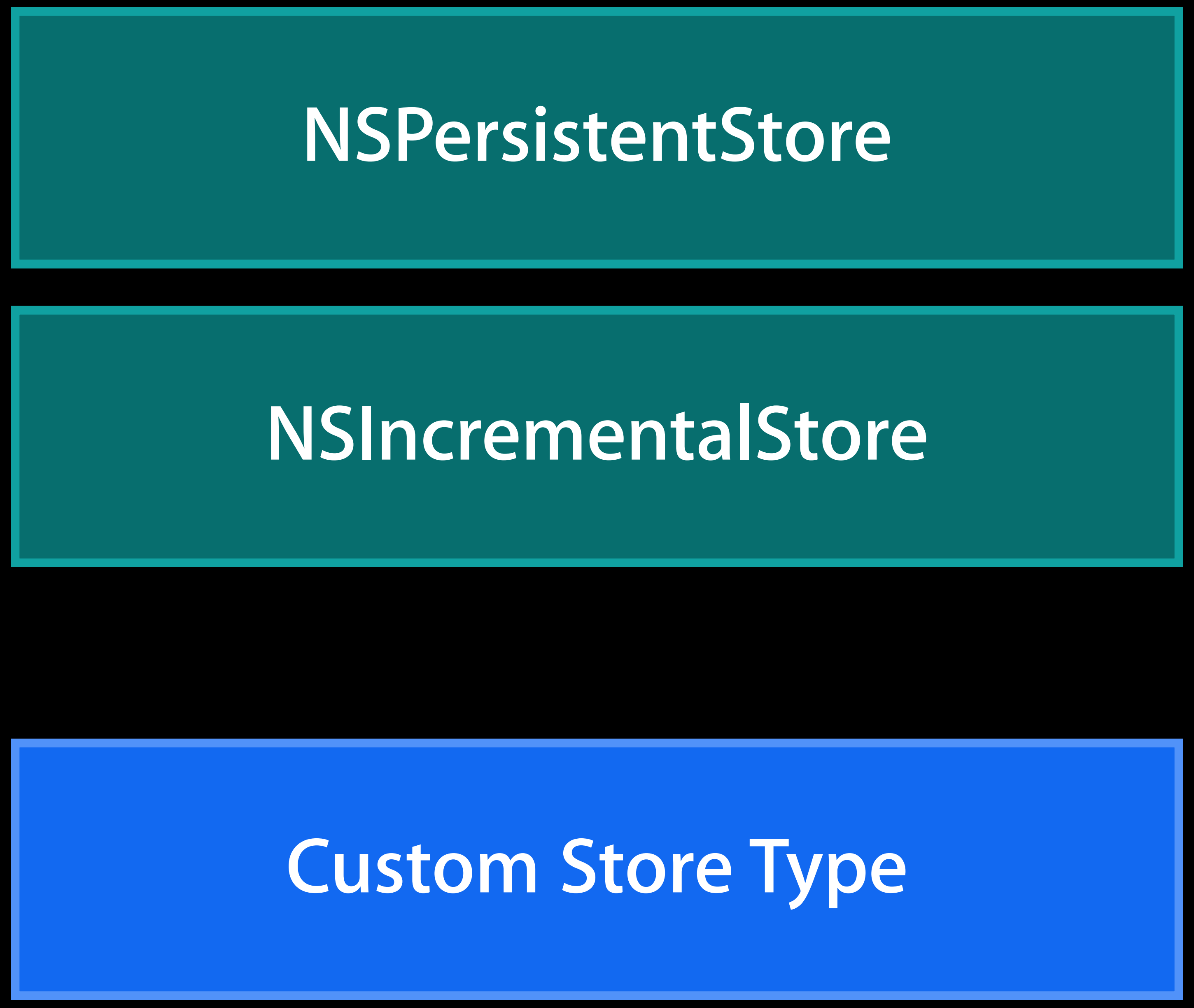
- Reduce cross-entity fetches

```
fetch.relationshipKeyPathsForPrefetching = @[ @"artist", @"catalog" ];
```

Core Data

NSIncrementalStore

Tip
26



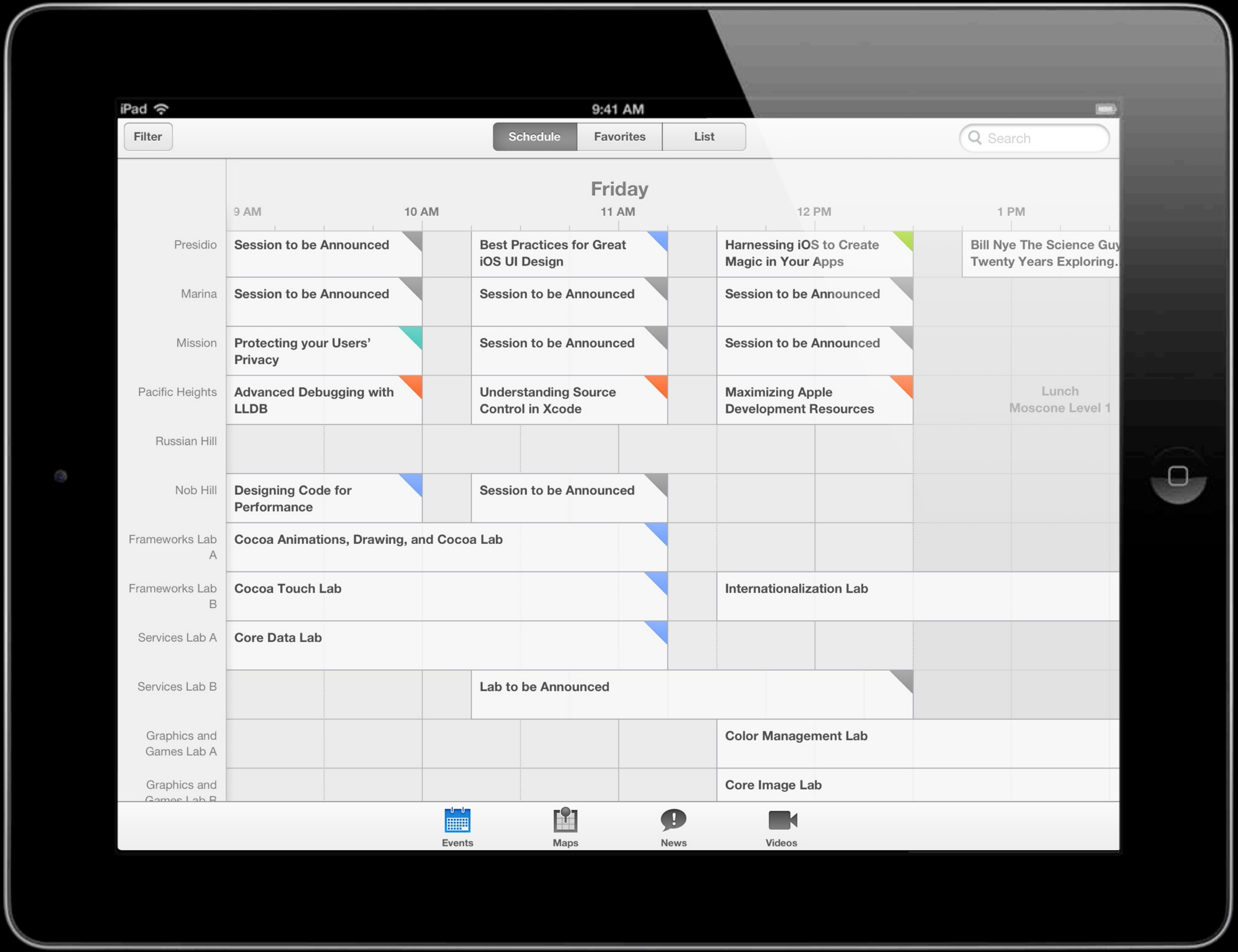
UIKit

Implementation tips

UIKit

UICollectionView

Tip 27



Friday

	9 AM	10 AM	11 AM	12 PM	1 PM
Presidio	Session to be Announced	Best Practices for Great iOS UI Design	Harnessing iOS to Create Magic in Your Apps	Bill Nye The Science Guy Twenty Years Exploring.	
Marina	Session to be Announced	Session to be Announced	Session to be Announced		
Mission	Protecting your Users' Privacy	Session to be Announced	Session to be Announced		
Pacific Heights	Advanced Debugging with LLDB	Understanding Source Control in Xcode	Maximizing Apple Development Resources	Lunch Moscone Level 1	
Russian Hill					
Nob Hill	Designing Code for Performance	Session to be Announced			
Frameworks Lab A	Cocoa Animations, Drawing, and Cocoa Lab				
Frameworks Lab B	Cocoa Touch Lab		Internationalization Lab		
Services Lab A	Core Data Lab				
Services Lab B		Lab to be Announced			
Graphics and Games Lab A			Color Management Lab		
Graphics and Games Lab B			Core Image Lab		

UIKit

UICollectionView

Tip

27

UIKit

UICollectionView

Tip 27

- Grid views, custom layouts
- Similar to UITableView

```
- (NSInteger)numberOfSectionsInCollectionView:(UICollectionView *)cv {
    return self.collectionViewSections.count;
}

- (NSInteger)collectionView:(UICollectionView *)cv
             numberOfItemsInSection:(NSInteger)section {
    return [[self sessionsInSection:section] count];
}
```

UIKit

UICollectionView

Tip

27

UIKit

UICollectionView

Tip

27

```
- (UICollectionViewCell *)collectionView:(UICollectionView *)cv
                  cellForItemAtIndexPath:(NSIndexPath *)path {

    NSArray *sessions = [self sessionsInSection:path.section];
    WWDCSession* session = sessions[indexPath.item];

    WWDCiPadScheduleSessionView *cell = nil;
    cell = [cv dequeueReusableCellWithReuseIdentifier:@"sessionView"
                                         forIndexPath:path];
    [cell setSession:session];
    return cell;
}
```

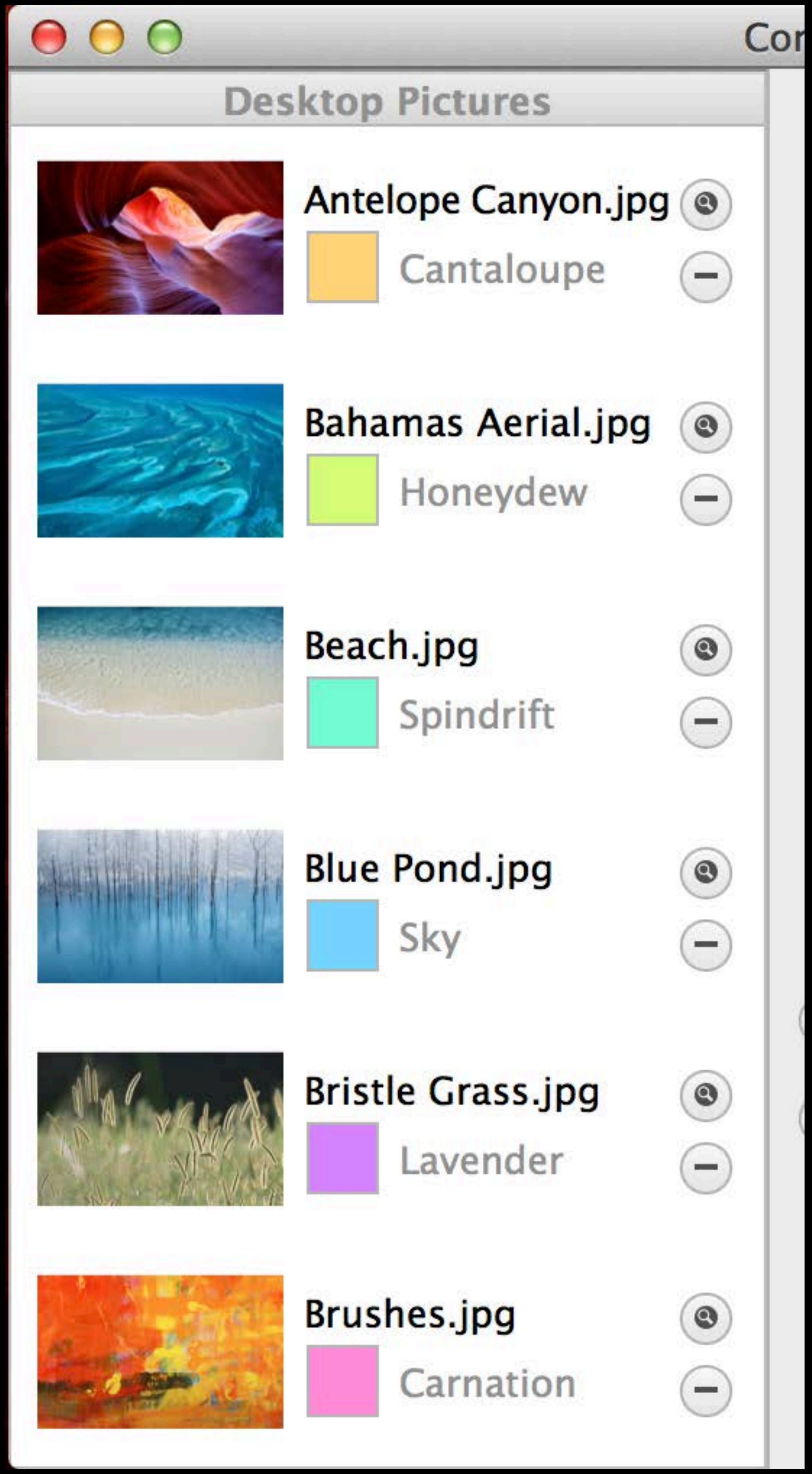
AppKit

Modern improvements

AppKit

View-based NSTableView

Tip 28



Desktop Pictures

Antelope Canyon.jpg	🟧	Cantaloupe
Bahamas Aerial.jpg	🟩	Honeydew
Beach.jpg	🟩	Spindrift
Blue Pond.jpg	🟦	Sky
Bristle Grass.jpg	🟪	Lavender
Brushes.jpg	🟪	Carnation

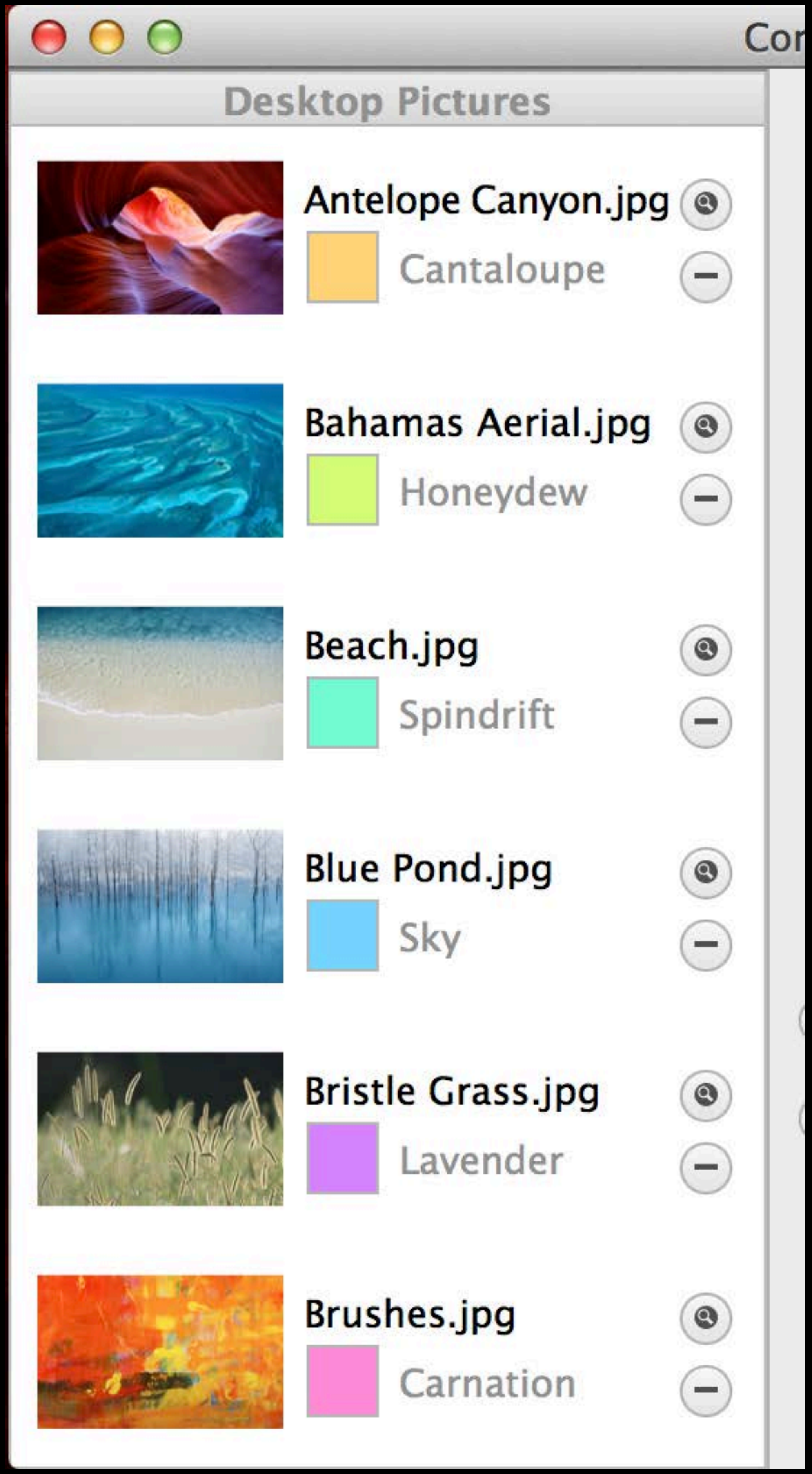
AppKit

View-based NSTableView

Tip 28

- Similar to UITableView
- Hardware accelerated
- Arbitrary views, no NSCell subclassing!
- Layout in IB
- Animation

```
NSTableViewAnimationEffectFade
NSTableViewAnimationEffectGap
NSTableViewAnimationSlideUp
NSTableViewAnimationSlideDown
NSTableViewAnimationSlideLeft
NSTableViewAnimationSlideRight
```



AppKit

View-based tables

Tip

29

```
- (NSInteger)numberOfRowsInTableView:(NSTableView *)tableView {
    return self.objects.count;
}

- (NSTableRowView *)tableView:(NSTableView *)table
                             rowViewForRow:(NSInteger)row {
    MyRowView *rowView;
  rowView = [table makeViewWithIdentifier:@"RowView" owner:self];
    return rowView;
}
```

AppKit

View-based tables

Tip

29

```
- (NSView *)tableView:(NSTableView *)table
     viewForTableColumn:(NSTableColumn *)col row:(NSInteger)row {

    NSTableCellView *cellView = nil;
    id objectForRow = self.objects[row];
    BOOL isSelected = (row == self.selectedRow);

    if ([self tableView:table isGroupRow:row]) {
        cellView = [table makeViewWithIdentifier:@"GroupView" owner:self];
        cellView.textField.stringValue = [object title];
        cellView.textField.textColor = (isSelected) ? [NSColor blueColor];
        return cellView;
    } else {
        ...
```

AppKit

View-based tables

Tip

29

```
if ([self tableView:table isGroupRow:row]) {
    cellView = [table makeViewWithIdentifier:@"GroupView" owner:self];
    cellView.textField.stringValue = [object title];
    cellView.textField.textColor = (isSelected) ? [NSColor blueColor];
    return cellView;
} else {
    MyGridView *gridView = [table makeViewWithIdentifier:@"Grid"
                                                       owner:self];
    gridView.dataSource = self.gridDataSource;
    gridView.delegate = self;
    return gridView;
}
```

More Information

Dave DeLong
App Frameworks and Developer Tools Evangelist
delong@apple.com

Apple Developer Documentation
http://developer.apple.com/documentation

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Designing Code for Performance	Nob Hill Friday 9:00AM
Core Data Performance Optimization and Debugging	Nob Hill Wednesday 2:00PM

WWDC2013